22/12/2023

Compte Rendu de la SAÉ-501
Concevoir, réaliser et présenter une solution technique

Encadré par :
Alexis Charton, WorldSkills France's Regional Expert
Jean-Michel Bouillet , WorldSkills France's Regional Expert

Réseaux et Télécoms
IUT Nord Franche-Comté

Thomas Pinot
Vincent Bardot
Thomas Mirbey
Alexandre Berot-Armand
Arnaud Kastner
Kyllian Cuevas
Maxence Bitschine

Sommaire

1. Contexte

Cette SAÉ 5.01, intitulée “Concevoir, réaliser et présenter une solution technique”, a été encadrée par M.BOUILLET et M.CHARTON. Nous avons réalisé ce projet à l’IUT de Montbéliard du lundi 11 décembre au vendredi 22 décembre.

Le projet visait à mettre en place une infrastructure complète système et réseau basée sur des équipements réseaux réels et plusieurs serveurs de machines virtuelles (Proxmox VE et ESXi)

Chaque membre a pu apporter ses connaissances et son expertise pour concevoir une solution intégrée, depuis la phase de conception jusqu'à la réalisation finale.

Il était important de bien organiser le projet afin de travailler en équipe de manière efficace.

2. Organisation

Durant ce projet, nous avons travailler en équipe de 7 personne comprenant des parcours Kyllian CUEVAS, Thomas MIRBEY du parcours IOM, Arnaud KASTNER et Maxence BITSCHINE du parcours PilPro et Thomas PINOT, Alexandre BEROT-ARMAND et Vincent BARDOT du parcours cyber.
Les PilPRo ont été en partis à charge de piloter ce projet, en particulier Maxence qui était principal superviseur et manager ce qui à permit à Arnaud d’intervenir seulement quand Maxence avait besoin d’aide et se concentrer pleinement sur le technique avec nos collègues du parcours IOM et Cyber qui se concentrent quant à eux au côté technique.

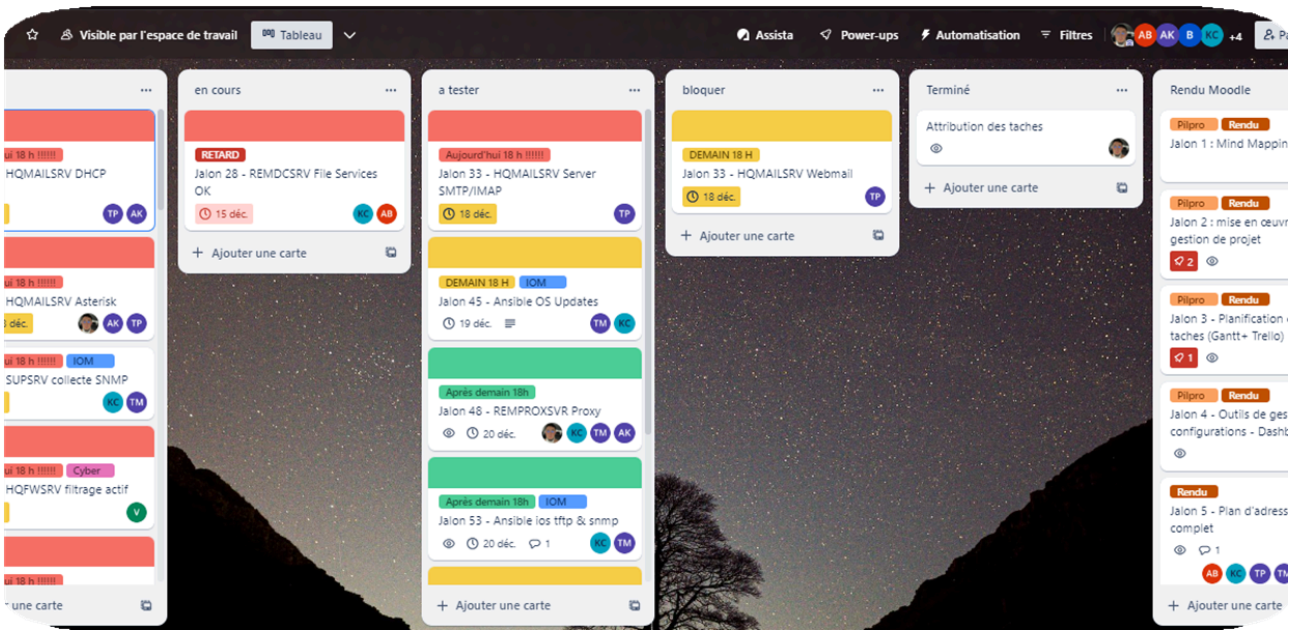
3. Gestion

a. Trello

Trello, un outil de gestion de projet en ligne, se distingue par sa simplicité et son approche visuelle. Organisé autour de tableaux virtuels, chaque projet est représenté par des listes et des cartes, permettant un suivi intuitif du flux de travail.

Les fonctionnalités incluent la collaboration en temps réel, des notifications instantanées, des intégrations flexibles avec d'autres outils, et une personnalisation poussée des cartes.

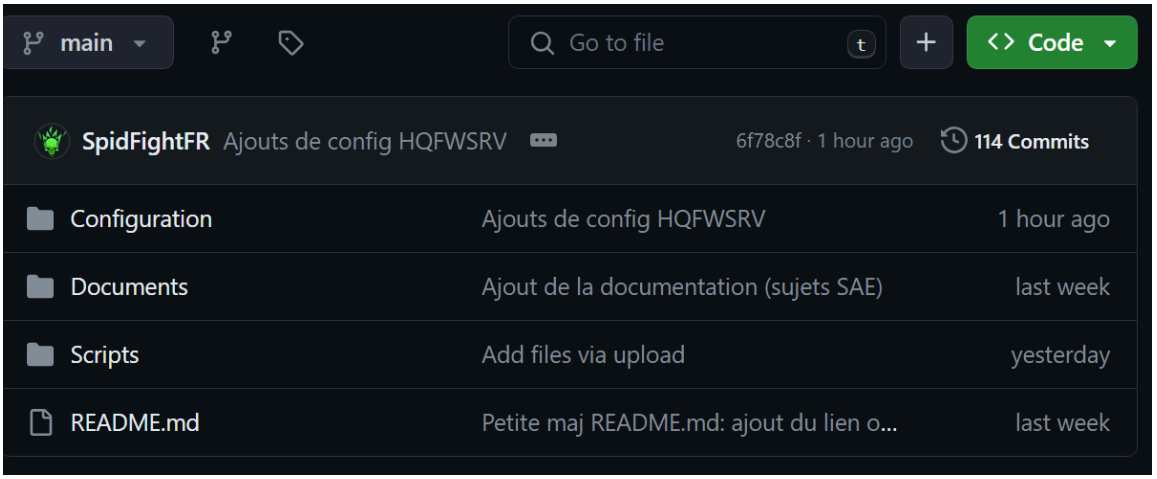
Avec une interface conviviale et une adaptabilité à toutes les échelles de projets, Trello offre une solution gratuite de base, rendant la gestion de tâches et de projets à la fois simple et accessible.



b. Github

GitHub est une plateforme de développement logiciel basée sur le Web qui facilite la gestion de projets informatiques, la collaboration et le contrôle de version. GitHub utilise Git, un système de contrôle de version distribué, pour suivre les changements dans le code source au fil du temps.

Ceci permet aux équipes de travailler ensemble sur des projets, quel que soit leur emplacement, en offrant des fonctionnalités telles que des branches pour travailler sur des fonctionnalités spécifiques. Nous avons donc utilisé Github pour faire du versionnage de nos configurations de routeurs, switchs, scripts, etc.



c. Plan d'adressage IP

Nous avons défini notre plan d'adressage IP grâce à la méthode VLSM (Variable Length Subnet Mask) afin d'optimiser la gestion des adresses IP et des masques de sous-réseau.

Le but était de d'optimiser le découpage des plages IP pour ne pas gâcher d'adresses.

Nous avons dû modifier plusieurs fois ce plan pour l'adapter aux différents changements que nous n'avions pas initialement prévus et que nous avons apportés au réseau, comme la mise en place de VIP entre nos routeurs.

Le plan d'adressage réseau se trouve dans les annexes.

4. Virtualisation

Étant donné la complexité du projet, nous ne pouvions pas nous reposer uniquement sur du matériel physique.

Nous avons donc utilisé plusieurs solutions de virtualisation, plus précisément, des hyperviseurs de tier 1. Les hyperviseurs de tier 1 sont caractérisés par le fait qu'ils fassent partie - ou soient, eux-même, le système d'exploitation.

a. Proxmox

Nous avons utilisé Proxmox, solution open source basée sur Debian-Linux.

Installé dans des serveurs fournis par l'université.
Ayant emprunté le proxmox 17 et 18, la partie INTERNET se situait dans le proxmox 18 et la partie Remote se situait dans le proxmox 17.

Le réseau remote était géré par le routeur REMFW (Cisco ISR1000) qui faisait le pont entre la carte VMBR1, une interface proxmox qui servait de switch entre les machines du réseau. L'interface VMBR0 faisait quant à elle le lien entre le routeur virtuel et le switch physique sur lequel nous pouvions connecter le cœur de réseau.

Pour le réseau Internet, toutes les machines étaient directement reliées à l'interface VMBR0, le reste se déroulait sur le WANRTR.

b. ESXi

Esxi en opposition à Proxmox est une solution propriétaire appartenant à l'entreprise VMWare.

C'est aussi un Hyperviseur de Tier 1, il est son propre système d'exploitation.
ESXi était installé sur deux ordinateurs fixes ayant 500Gb de stockage chacun.

Les hyperviseurs en question sont similaires, ils permettent la même chose quand bien même ils ne partagent pas le même vocabulaires et malgré les disparités en termes de configuration (VM Bridges vs VSwitches par exemple).

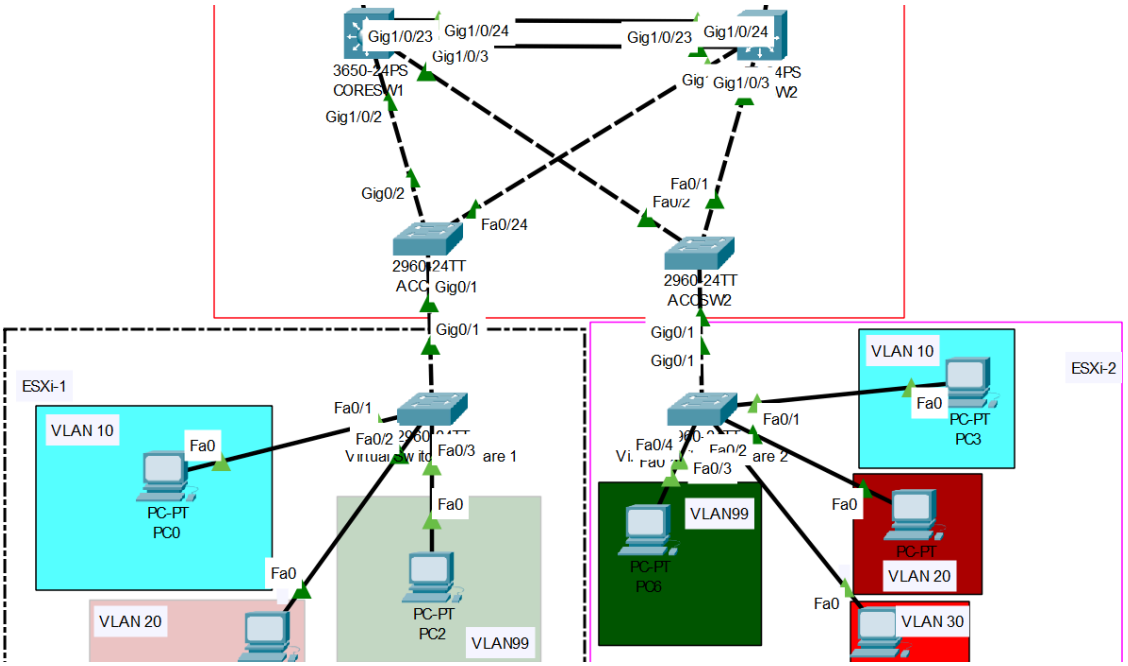
5. Solution mise en place

a. Coeur du réseau

i. Switching

Nous avons préparé l'ensemble du réseau Switching sur une simulation packet tracer pour pouvoir tester le bon fonctionnement de nos configurations.

Préparation du réseau en faisant une simulation sous packet Tracer :



1. EtherChannel

Sur le schéma, nous avons délimité les différents VLAN par des carrés de couleurs pour simplifier la configuration et les tests des clients. Les zones sont également répertoriées : cœur du réseau, ESXI-1 et ESXI-2.

Nous avons configuré un lien etherchannel entre CORESW1 et CORESW2. Un lien EtherChannel ou PortChannel permet de regrouper plusieurs liens dans un seul lien virtuel (link aggregation). Le but est d'augmenter la bande passante et de fournir de la redondance de liaisons.

```
interface FastEthernet1/0/23
 switchport trunk encapsulation dot1q
 switchport trunk native vlan 666
 switchport trunk allowed vlan 10,20,30,99,300
 switchport mode trunk
 channel-group 1 mode active
!
interface FastEthernet1/0/24
 switchport trunk encapsulation dot1q
 switchport trunk native vlan 666
 switchport trunk allowed vlan 10,20,30,99,300
 switchport mode trunk
 channel-group 1 mode active
```

Lorsque l'on souhaite mettre en place un portchannel, on doit configurer les différents ports qui appartiendront à la liaison. Ceux-ci seront regroupés dans la même liaison EtherChannel définie par un ID.

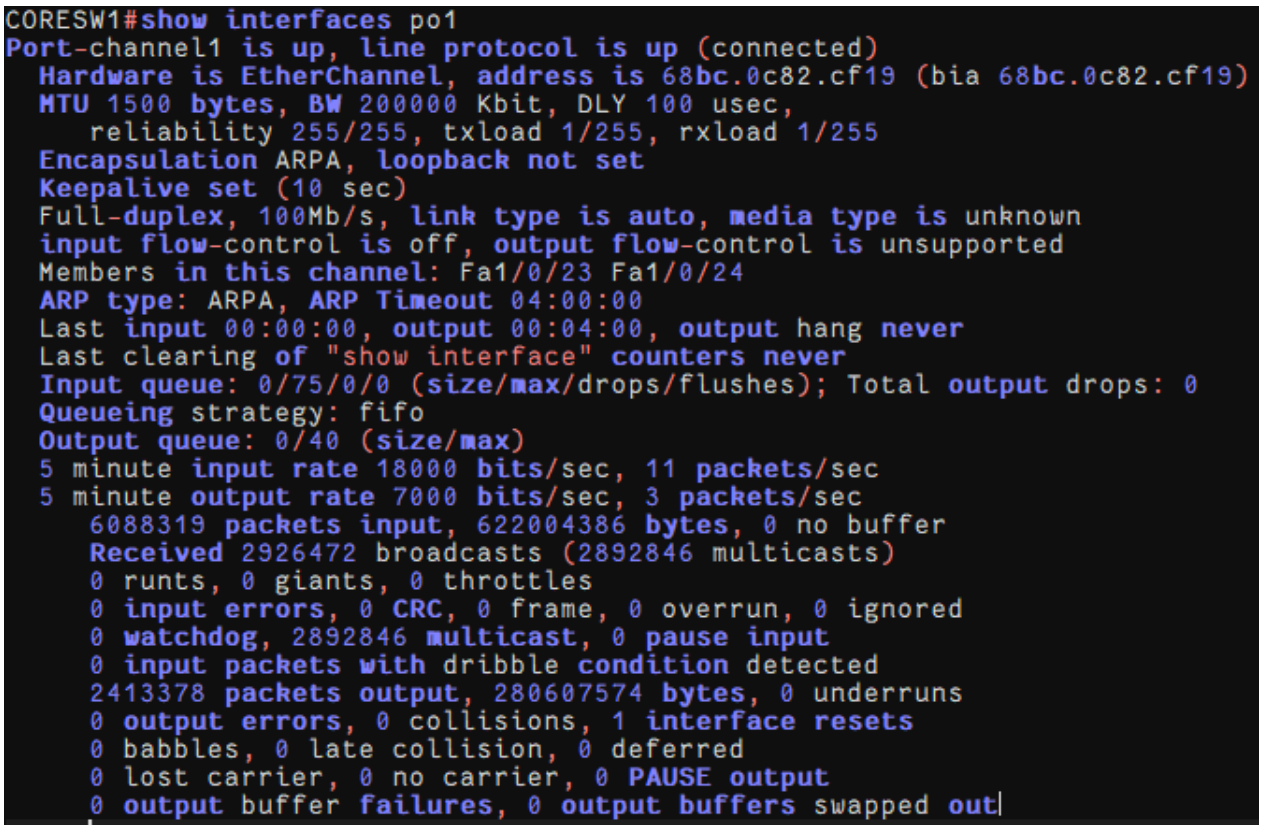
Si on souhaite faire des modifications sur l'agrégation de liens, on doit impérativement modifier l'interface PortChannel correspondant et non pas les liens y appartenant.

```
interface Port-channel1
 switchport trunk encapsulation dot1q
 switchport trunk native vlan 666
 switchport trunk allowed vlan 10,20,30,99,300
 switchport mode trunk
```

On peut contrôler l'état des PortChannel via la commande show EtherChannel. On retrouvera l'ensemble des agrégations de liens créés.

```
CORESW1#show etherchannel
                Channel-group listing:
                ----------------------

Group: 1
----------
Group state = L2
Ports: 2   Maxports = 16
Port-channels: 1 Max Port-channels = 16
Protocol:   LACP
Minimum Links: 0
```

On peut également récupérer plus d'informations sur le lien EtherChannel en affichant les informations de l'interface PortChannel.


CORESW1#show interfaces po1
Port-channel1 is up, line protocol is up (connected)
Hardware is EtherChannel, address is 68bc.0c82.cf19 (bia 68bc.0c82.cf19)
MTU 1500 bytes, BW 200000 Kbit, DLY 100 usec,
reliability 255/255, txload 1/255, rxload 1/255
Encapsulation ARPA, loopback not set
Keepalive set (10 sec)
Full-duplex, 100Mb/s, link type is auto, media type is unknown
input flow-control is off, output flow-control is unsupported
Members in this channel: Fa1/0/23 Fa1/0/24
ARP type: ARPA, ARP Timeout 04:00:00
Last input 00:00:00, output 00:04:00, output hang never
Last clearing of "show interface" counters never
Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
Queueing strategy: fifo
Output queue: 0/40 (size/max)
5 minute input rate 18000 bits/sec, 11 packets/sec
5 minute output rate 7000 bits/sec, 3 packets/sec
6088319 packets input, 622004386 bytes, 0 no buffer
Received 2926472 broadcasts (2892846 multicasts)
0 runts, 0 giants, 0 throttles
0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
0 watchdog, 2892846 multicast, 0 pause input
0 input packets with dribble condition detected
2413378 packets output, 280607574 bytes, 0 underruns
0 output errors, 0 collisions, 1 interface resets
0 babbles, 0 late collision, 0 deferred
0 lost carrier, 0 no carrier, 0 PAUSE output
0 output buffer failures, 0 output buffers swapped out

2. VTP

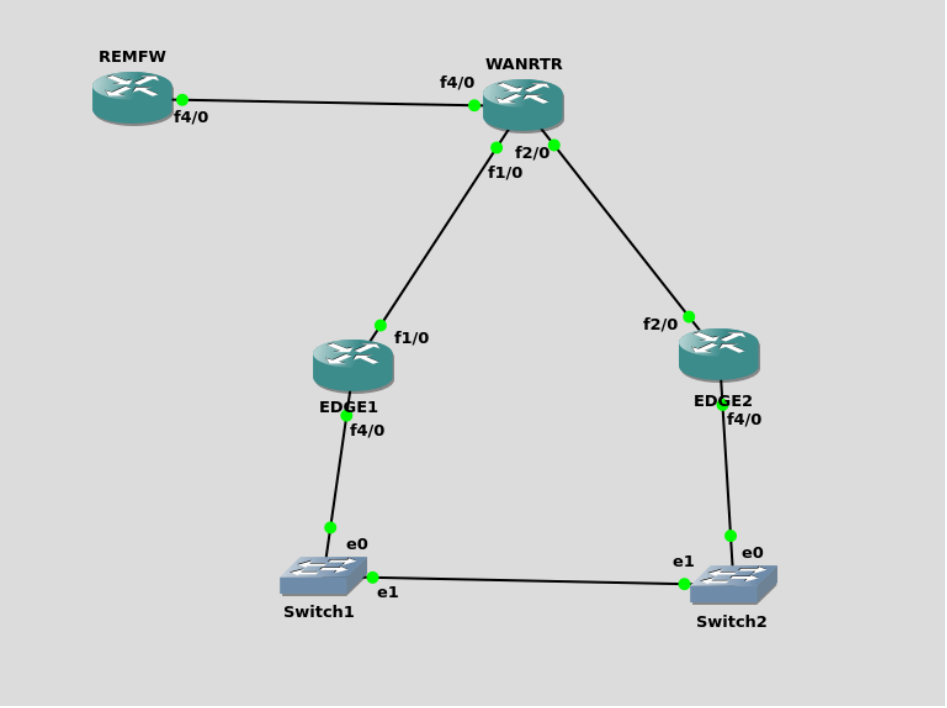
Le protocole VLAN TRUNK PROTOCOL (VTP) permet de propager la configuration des VLANs depuis un switch en mode serveur sur des switchs en mode clients.

```
CORESW1#sh vtp status
VTP Version capable             : 1 to 3
VTP version running             : 2
VTP Domain Name                 : wsl2024.org
VTP Pruning Mode                : Disabled
VTP Traps Generation            : Disabled
Device ID                       : 68bc.0c82.cf00
Configuration last modified by 0.0.0.0 at 3-1-93 01:06:23
Local updater ID is 10.1.10.60 on interface Vl10 (lowest numbered VLAN interface found)

Feature VLAN:
--------------
VTP Operating Mode              : Server
Maximum VLANs supported locally : 1005
Number of existing VLANs        : 20
Configuration Revision          : 9
MD5 digest                      : 0x80 0x6F 0x10 0x09 0x44 0x16 0x64 0x12
                                  0x43 0x82 0x19 0x2C 0x28 0xBA 0xAE 0x5A
```

ii. Routing

Avant de configurer les routeurs physiques, nous avons déjà réalisé une première maquette du réseau sur GNS3. Celle-ci nous a permis de vérifier si tout était fonctionnel. L'intérêt de faire une configuration de test comme celle-ci permet de plus facilement identifier et isoler les problèmes qui pourraient arriver lors du déploiement des configurations.

Nous avons utilisé cette configuration que nous avons mise sur tous nos routeurs.



1. Sous interfaces Dot1Q

L'encapsulation dot1q avec des sous-interfaces permet de segmenter le trafic en fonction des VLAN. Chaque sous-interface correspond à un VLAN spécifique. Cela permet de segmenter le réseau dans différents VLAN.

Dans notre cas, chaque routeur possédait un numéro de vlan associé à un type de trafic (Internet ou local).

Device	VLAN ID	Description
EDGE1	13	Used for MAN connection
	14	Used for INET connection
EDGE2	15	Used for MAN connection
	16	Used for INET connection

```
interface GigabitEthernet0/0.13
 description WANRTR VRF MAN
 encapsulation dot1Q 13
 ip address 10.1.254.253 255.255.255.252

interface GigabitEthernet0/0.14
 description WANRTR VRF INET
 encapsulation dot1Q 14
 ip address 91.1.222.98 255.255.255.252
```

Pour configurer une sous interface, il ne faut pas configurer l'interface physique du routeur.

Une sous interface se renseigne en ajoutant .numero à la fin d'une interface physique.

Il faut ensuite ajouter le tag du VLAN avec la commande `encapsulation dot1Q numero_vlan`.

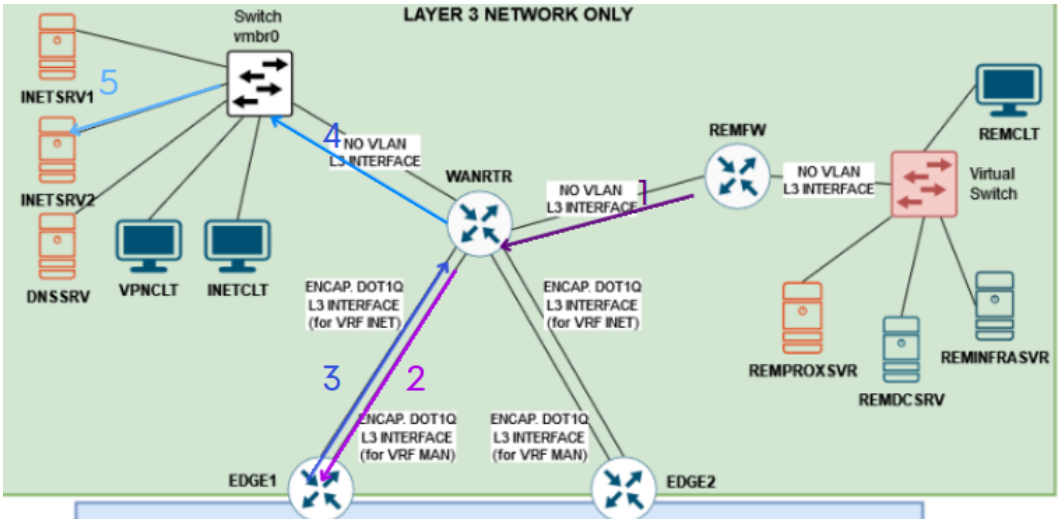
2. VRF

Les VRF (Virtual Routing and Forwarding) sont des instances virtuelles de routeurs qui servent à segmenter le réseau. Elles permettent de créer des réseaux virtuels avec leurs propres tables de routage indépendantes les unes des autres. Nous avons créé deux VRF, le but était de séparer le trafic d'Internet (INET) du réseau local (MAN) sur notre routeur WANRTR.

```
Routing Table: MAN

Gateway of last resort is 217.1.160.6 to network 0.0.0.0

S*    0.0.0.0/0 [1/0] via 217.1.160.6
      10.0.0.0/8 is variably subnetted, 9 subnets, 4 masks
O E2     10.1.0.0/16
           [110/20] via 10.1.254.249, 16:08:32, GigabitEthernet0/0.15
O        10.1.254.240/30
           [110/2] via 10.1.254.249, 16:08:32, GigabitEthernet0/0.15
O        10.1.254.244/30
           [110/2] via 10.1.254.253, 15:14:56, GigabitEthernet0/1.13
C        10.1.254.248/30 is directly connected, GigabitEthernet0/0.15
L        10.1.254.250/32 is directly connected, GigabitEthernet0/0.15

Routing Table: INET

Gateway of last resort is 217.1.160.6 to network 0.0.0.0

S*    0.0.0.0/0 [1/0] via 217.1.160.6
      8.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C        8.8.1.0/28 is directly connected, FastEthernet0/1/0
L        8.8.1.14/32 is directly connected, FastEthernet0/1/0
      10.0.0.0/8 is variably subnetted, 6 subnets, 2 masks
      31.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
B        191.15.157.32 [20/0] via 31.1.126.14, 15:18:03
      217.1.160.0/29 is subnetted, 1 subnets
B        217.1.160.0 [20/0] via 31.1.126.14, 15:18:03
```

On peut voir deux tables de routage distinctes pour chaque VRF. Il n'est pas possible de passer directement d'une table à l'autre directement.

Pour passer d'un VRF à l'autre, le trafic devait être envoyé vers un autre routeur ne possédant pas de VRF qui lui renvoyait ensuite les données sur le bon VRF conformément au schéma suivant.



Pour permettre ce renvoie, nous avons configuré une route par défaut par VRF sur l'adresse IP HSRP des routeurs EDGE.

```
ip route vrf INET 0.0.0.0 0.0.0.0 217.1.160.6
ip route vrf MAN 0.0.0.0 0.0.0.0 217.1.160.6
```

Dans le cadre de cette Saé, chaque VRF utilisait son propre protocole de routage. OSPF pour MAN et BGP pour INET.

3. OSPF

Lors de la configuration d'OSPF, nous avons pu voir qu'il y avait une subtilité au niveau des VRF. En effet, lorsque l'on souhaite utiliser OSPF avec un VRF, on doit le spécifier au niveau de la déclaration du processus OSPF.

```
router ospf 1 vrf MAN
 router-id 0.0.0.3
 redistribute connected
 redistribute static
 network 10.1.254.248 0.0.0.3 area 1
 network 10.1.254.252 0.0.0.3 area 1
 network 10.116.1.0 0.0.0.3 area 1
```

Configuration d'OSPF avec VRF

```
router ospf 1
 router-id 0.0.0.1
 redistribute connected
 redistribute static
 redistribute bgp 65116
 network 10.1.254.244 0.0.0.3 area 1
 network 10.1.254.252 0.0.0.3 area 1
```

Configuration d'OSPF sans VRF

Pour contrôler la configuration d'OSPF, nous avons contrôlé les tables de routage de nos routeurs ainsi que les voisins OSPF.

```
EDGE1#sh ip ospf nei

Neighbor ID     Pri   State           Dead Time   Address         Interface
0.0.0.3           1   FULL/DR         00:00:39    10.1.254.254    GigabitEthernet0/0.13
0.0.0.4           1   FULL/DR         00:00:36    10.1.254.246    GigabitEthernet0/1.100
```

4. BGP

La configuration de BGP est celle qui nous a posé le plus de problèmes. De la même manière que pour OSPF, lorsque l'on souhaite configurer BGP, il faut préciser que ce processus sera exécuté sur un VRF via une address-family IPV4.

```
router bgp 65130
 bgp router-id 8.8.8.1
 bgp log-neighbor-changes
 !
 address-family ipv4 vrf INET
  network 31.1.126.12 mask 255.255.255.252
  network 91.1.222.96 mask 255.255.255.252
  redistribute connected
  redistribute static
  neighbor 31.1.126.14 remote-as 65116
  neighbor 31.1.126.14 activate
  neighbor 91.1.222.98 remote-as 65116
  neighbor 91.1.222.98 activate
 exit-address-family
```

Configuration de BGP avec VRF

```
router bgp 65116
 bgp router-id 91.1.126.98
 bgp log-neighbor-changes
 neighbor 91.1.222.97 remote-as 65130
 neighbor 217.1.160.5 remote-as 65116
 !
 address-family ipv4
  bgp redistribute-internal
  network 91.1.222.96 mask 255.255.255.252
  network 217.1.160.0 mask 255.255.255.248
  redistribute connected
  redistribute static
  redistribute ospf 1
  neighbor 91.1.222.97 activate
  neighbor 217.1.160.5 activate
 exit-address-family
```

Configuration de BGP sans VRF

Une fois cette partie faite sur la maquette, le réseau était fonctionnel donc nous l'avons mis en place sur les routeurs physiques. Lors de la réalisation de nos tests, nous nous sommes rendus compte que nos voisins BGP ne montaient pas sur une liaison bien spécifique.

Nous n'avions pas eu ce problème lors de la configuration GNS3 et notre configuration était similaire à une autre liaison qui fonctionnait. Après avoir validé la bonne configuration avec Alexis Charton et l'avoir recommencée plusieurs fois, nous avons changé de routeur et la configuration a fonctionné.

```
Neighbor        V           AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
91.1.222.97     4        65130    1021    1021      688    0    0 15:15:32        3
217.1.160.5     4        65116    1023    1017      688    0    0 15:14:54       10
```

5. NAT

```
ip access-list extended NAT-ACL
 permit ip any any
```

Cette configuration nous a également posé plusieurs problèmes. En effet, pour simplifier la phase de test, nous avons créé une access-list pour notre NAT permettant à n'importe quelle adresse IP d'un réseau d'être translaté.

Les annonces de voisins BGP étaient donc translatées et les voisins n'avaient donc pas à se connecter. Nous avons pu vérifier cette information avec la commande `debug bgp all` et en regardant l'état des voisins BGP.

```
EDGE1#sh ip bgp all sum
Neighbor        V           AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
91.1.222.97     4        65130       0       0        1    0    0   Never       Idle
217.1.160.5     4        65116    1023    1017      688    0    0 15:14:54       10
```

En modifiant l'access-list et en refusant explicitement l'accès à la NAT à notre voisin BGP, le problème a été réglé.

```
ip access-list extended NAT-ACL
 deny 91.1.222.97 0.0.0.0 any
 permit ip 10.1.0.0 0.0.255.255 any
```

Pour configurer une access-list, il faut lier l'access-list à une interface. via la commande ip nat inside permettant de préciser les réseaux à nater. Dans notre cas, on nate tous nos réseaux privés en 10.1.0.0/16 vers l'adresse IP de notre interface GigabitEthernet0/0.14.

```
ip nat inside source list NAT-ACL interface GigabitEthernet0/0.14 overload


interface GigabitEthernet0/0.13
 description WANRTR VRF MAN
 encapsulation dot1Q 13
 ip address 10.1.254.253 255.255.255.252
 ip nat inside
 ip virtual-reassembly in
!
interface GigabitEthernet0/0.14
 description WANRTR VRF INET
 encapsulation dot1Q 14
 ip address 91.1.222.98 255.255.255.252
 ip nat outside
 ip virtual-reassembly out
```

On peut contrôler les translations d'adresses avec deux commandes :

- `show ip nat statistics`
- `show ip nat translations`

La capture d'écran suivante vous permettra de voir une partie de translations d'adresses.

```
EDGE1#sh ip nat translations
Pro Inside global      Inside local       Outside local      Outside global
icmp 91.1.222.98:27482 10.1.99.1:27482    8.8.1.3:27482      8.8.1.3:27482
icmp 91.1.222.98:27482 10.1.99.1:27482    8.8.1.12:27482     8.8.1.12:27482
icmp 91.1.222.98:27484 10.1.99.1:27484    8.8.1.13:27484     8.8.1.13:27484
udp 91.1.222.98:32924  10.1.99.1:32924    8.8.1.13:161       8.8.1.13:161
udp 91.1.222.98:37672  10.1.99.1:37672    8.8.1.12:161       8.8.1.12:161
udp 91.1.222.98:39573  10.1.99.1:39573    8.8.1.13:161       8.8.1.13:161
udp 91.1.222.98:41401  10.1.99.1:41401    8.8.1.13:161       8.8.1.13:161
udp 91.1.222.98:46918  10.1.99.1:46918    8.8.1.12:161       8.8.1.12:161
udp 91.1.222.98:51223  10.1.99.1:51223    8.8.1.3:161        8.8.1.3:161
udp 91.1.222.98:52026  10.1.99.1:52026    8.8.1.3:161        8.8.1.3:161
udp 91.1.222.98:55503  10.1.99.1:55503    8.8.1.12:161       8.8.1.12:161
tcp 91.1.222.98:46120  10.1.99.9:46120    162.159.128.233:443 162.159.128.233:443
tcp 91.1.222.98:48396  10.1.99.9:48396    162.159.128.233:443 162.159.128.233:443
tcp 91.1.222.98:48404  10.1.99.9:48404    162.159.137.232:443 162.159.137.232:443
tcp 91.1.222.98:48430  10.1.99.9:48430    162.159.137.232:443 162.159.137.232:443
```

b. Services

i. Windows

Les serveurs Windows étaient au cœur de l'infrastructure du système d'information de la SAE. HQDCSRV était le point central des clients Windows dans HQ et la partie REMOTE était gérée par les serveurs REMDCSRV et REMINFRASRV.

Dans le cadre de l'administration de ces serveurs nous avons dû utiliser les services proposés par le gestionnaire de serveur Windows.

Une grande partie de ces services ont été configurés via des scripts en PowerShell.

1. Dynamic Host Configuration Protocle (DHCP)

Uniquement présent sur REMDCSRV et REMINFRASRV, l'entièreté du DHCP a été configurée via un script PowerShell (Visible en annexe).

Le script génère une étendue DHCP entre 10.1.100.45 et 10.1.100.125 avec une durée de bail de 2 heures.

Il ajoute aussi une route par défaut vers le routeur REMFW (10.1.100.126) et l'adresse du serveur DNS principale REMDCSRV (10.1.100.1).

Pour finir, le script active une relation de FailOver avec LoadBalancing 50/50 avec le serveur REMINFRASRV. Il y a le

L'activation du serveur NTP vers HQINFRASRV n'était pas configurée par manque de temps pour tester son fonctionnement.

2. Domain Name Server (DNS)

Uniquement présent sur REMDCSRV et REMINFRASRV, l'entièreté du DHCP a été configurée via un script PowerShell (Visible en annexe).

Le script génère une étendue DHCP entre 10.1.100.45 et 10.1.100.125 avec une durée de bail de 2 heures.

Il ajoute aussi une route par défaut vers le routeur REMFW (10.1.100.126) et l'adresse du serveur DNS principale REMDCSRV (10.1.100.1).

Pour finir, le script active une relation de FailOver avec Loadbalancing 50/50 avec le serveur REMINFRASRV. Il y a l'activation du serveur NTP vers HQINFRASRV n'était pas configurée par manque de temps pour tester son fonctionnement.

3. Active Directory Domain Services (ADDS)

L'Active Directory a été généré entièrement avec des scripts en PowerShell en utilisant des fichiers CSV. Il se trouve aussi dans les annexes.

Le script génère les différentes unités d'organisation imbriquées (différentes en fonction du serveur) puis extrait les informations des CSV et remplis les profils des différents utilisateurs avec les informations extraites.

Les scripts créer aussi des scripts en Batch pour donner les NetShare aux utilisateurs (utilisable en backup en cas de problème avec les GPO prévu pour ça).

Il y a eu plusieurs erreurs (notamment avec les informations utilisées par les scripts) dues à des mauvais choix dans la réalisation des scripts.

4. Active Directory Certificate Services (ADCS)

L'AD CS est le service de certificats intégré à Windows.

Ce service était demandé mais nous n'avons pas eu le temps de le mettre en place avec le temps imparti.

5. Files Services

Le service de fichier a été géré en partie avec le script de l'Active Directory, en partie manuellement et en partie par GPO.

La partie sur les GPO sera développé dans le chapitre des GPO

Le script AD devait générer les dossiers des utilisateurs et des services ainsi que l'attribution des droits sur les dossiers.

Les dossiers contenant les dossiers créer par le script ont été créés à la main et rien

Il y a eu un problème sur les droits d'attributions des dossiers créés manuellement qui ont limité les droits des utilisateurs du script.

Ce problème a donc empêché les utilisateurs de pouvoir créer ou modifier des dossiers dans les dossiers où ils avaient ces droits.

Ce problème aurait pu être réglé en changeant les droits sur les fichiers de bases (ou simplement en les créant avec un script et en les paramétrant en même temps.

6. Distributed Files System (DFS)

Le DFS a été configuré pour permettre la haute disponibilité des dossiers des utilisateurs et des services sur le site REMOTE.

La mise en place aurait pu être faite par script mais n'ayant pas les connaissances pour le faire, le service a été configuré manuellement plutôt rapidement.

Le service a été fonctionnel directement et sans problème.

7. Group Policy (GPO)

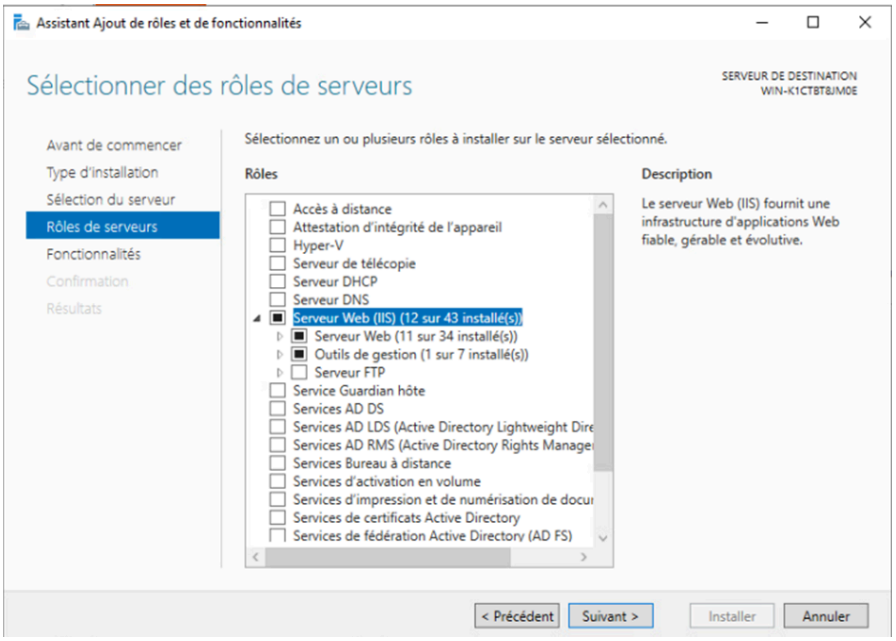
Les GPO ont été faites sur les serveurs ADDS. Elles avaient différentes utilisations comme le mappage de lecteur réseau, l'interdiction d'ouvrir le panneau de configuration ou de mettre un fond d'écran personnalisé.

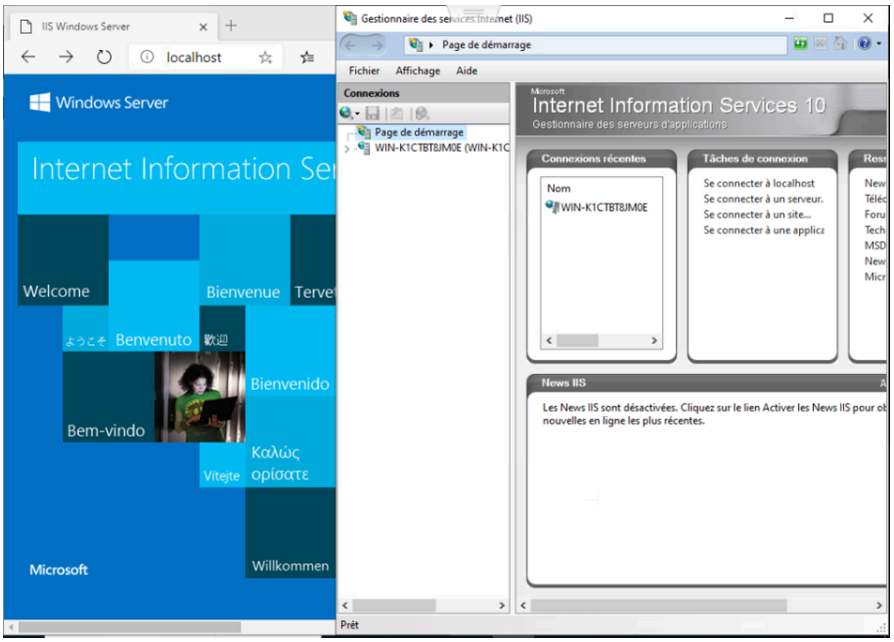
Dans l'ensemble, certaines ont marché, d'autres non. Le principal problème a été le non fonctionnement de certaines GPO durant la démonstration finale qui étaient fonctionnelles sur nos tests personnels pendant les jours d'avant sans changement dessus.

Un deuxième problème a été le manque de temps, ce qui a empêché la mise en place d'une partie de la sécurité tel que les GPO.

8. IIS


Tableau de bord
Serveur local
Tous les serveurs
IIS
Services de fichiers et d...
BIENVENUE DANS GESTIONNAIRE DE SERVEUR
DÉMARRAGE RAPIDE
NOUVEAUTÉS
EN SAVOIR PLUS
1 Configurer ce serveur local
2 Ajouter des rôles et des fonctionnalités
3 Ajouter d'autres serveurs à gérer
4 Créer un groupe de serveurs
5 Connecter ce serveur aux services cloud
Masquer


Assistant Ajout de rôles et de fonctionnalités
Sélectionner des rôles de serveurs
SERVEUR DE DESTINATION
WIN-K1CTBT8JM0E
Avant de commencer
Type d'installation
Sélection du serveur
Rôles de serveurs
Fonctionnalités
Confirmation
Résultats
Sélectionnez un ou plusieurs rôles à installer sur le serveur sélectionné.
Rôles
Accès à distance
Attestation d'intégrité de l'appareil
Hyper-V
Serveur de télécopie
Serveur DHCP
Serveur DNS
Serveur Web (IIS) (12 sur 43 installé(s))
Serveur Web (11 sur 34 installé(s))
Outils de gestion (1 sur 7 installé(s))
Serveur FTP
Service Guardian hôte
Services AD DS
Services AD LDS (Active Directory Lightweight Dire
Services AD RMS (Active Directory Rights Manage
Services Bureau à distance
Services d'activation en volume
Services d'impression et de numérisation de docu
Services de certificats Active Directory
Services de fédération Active Directory (AD FS)
Description
Le serveur Web (IIS) fournit une infrastructure d'applications Web fiable, gérable et évolutive.
< Précédent
Suivant >
Installer
Annuler


IIS Windows Server
localhost
Windows Server
Internet Information Ser
Welcome
Bienvenue
Benvenuto
Bienvenido
Bem-vindo
Καλώς ορίσατε
Vítejte
Willkommen
Microsoft
Gestionnaire des services Internet (IIS)
Page de démarrage
Fichier
Affichage
Aide
Connexions
Page de démarrage
WIN-K1CTBT8JM0E (WIN-K1C
Microsoft
Internet Information Services 10
Gestionnaire des serveurs d'applications
Connexions récentes
Nom
WIN-K1CTBT8JM0E
Tâches de connexion
Se connecter à localhost
Se connecter à un serveur.
Se connecter à un site...
Se connecter à une applica
News IIS
Les News IIS sont désactivées. Cliquez sur le lien Activer les News IIS pour ob
nouvelles en ligne les plus récentes.
Prêt

9. Script

Les scripts ont été particulièrement utiles dans ce projet. Lors d'un grave problème avec l'AD de REMDCSRV, un nouveau serveur a dû être déployé à la place ce qui a corrompu le DNS de HQDCSRV et de REMINFRASRV (à cause du Serial Number qui avait changé). Cette situation nous a forcé à refaire les trois serveurs ce qui a été considérablement accélérée par l'utilisation de script.

L'ensemble des scripts utilisés dans le projet sont trouvables dans les annexes

ii. Linux

1. ISC DHCP

La première chose que nous avons réalisée sur le serveur HQINFRASRV est de configurer le serveur DHCP, pour cela nous avons choisi la solution suivante : isc-dhcp-server.

Il devait correspondre à plusieurs exigences :

-Subnet : 10.N.20.X
- Netmask : To be defined
- Range : To be defined
- Gateway : To be defined
- Name server : hqdcsrv.hq.wsl2024.org
- Domain : hq.wsl2024.org
- NTP server : hqinfrasrv.wsl2024.org
- Lease : 2 hours

Ce qui nous a donné la configuration suivante :

```
subnet 10.1.20.0 netmask 255.255.254.0 {
        range 10.1.20.208 10.1.21.254;
        option routers 10.1.21.254;
        option domain-name-servers hqdcsrv.hq.wsl2024.org;
        option domain-name "hq.wsl2024.org";
        option ntp-servers hqinfrasrv.wsl2024.org;
        default-lease-time 7200;
        max-lease-time 7200;
}
```

Le serveur dhcp à une plage de 300 hôtes comme spécifié dans notre plan d'adressage.
Le lease time étant configuré en secondes, nous avons mis 7200 pour que cela corresponde aux 2 heures des consignes.

Nous n'avons pas spécialement rencontré de problème lors de la configuration de ce serveur dhcp.

2. Bind DNS

Sur le serveur DNSSRV, nous avons mis en place une installation de Bind9, un logiciel permettant d'héberger un serveur de nom de domaine.

On a fait une configuration avec deux bases de données en fonction des domaines wsl2024.org et worldskills.org.

Voici la configuration de notre serveur DNSSRV:

```
//Config bind custom, la config par défaut est dans /root/save/bind/named.conf.options
options {
        directory "/var/cache/bind";
        listen-on { 8.8.1.3; }; //ecouter les requettes sur la loopback et l'interface internet
        allow-query { any; };
        recursion no; //permet de faire des recherches de notre DNS vers d'autres DNS, A DESACTIVER SI LE DNS EST SUR INTERNET
        dnssec-validation no;
        //dnssec-enable yes;
        //dnssec-validation yes; //active le dnssec, on desactive pour l'instant
        //dnssec-lookaside auto;
        querylog yes;
};

zone "worldskills.org"{
        type master;
        file "/etc/bind/db.worldskills.org";
};

zone "wsl2024.org"{
        type master;
        file "/etc/bind/db.wsl2024.org";
};
```

Nous pouvons voir trois parties dans ce fichier config, la première détaille comment le serveur fonctionne, notamment sur quelle IP il doit écouter les requêtes, s'il doit enregistrer des logs des requêtes.

Les deux autres parties définissent où sont les fichiers databases contenant les entrées DNS des deux zones.

Voici le fichier wsl2024 :

```
;Fichier config zone wsl2024.org
;Tout ce qui suit un point virgule est un commentaire


$TTL 604800 ;Permet de maintenir le Time to live de notre zone a 604800 secondes, pendant 1 semaine,
@       IN      SOA     ns.wsl2024.org. admin.wsl2024.org. (
                        1               ;Serie
                        604800          ;TTL / Tmps de refresh data
                        86400           ;Retry
                        2419200         ;Expire
                        604800          ;Negative cache TTL
)

;Liste pour NameServers
                                IN      NS      ns.wsl2024.org.
ns                              IN      A       8.8.1.3

;Liste des Types A
hqfwsrv                         IN      A       217.1.160.1
vpn                             IN      A       191.5.157.33
webmail                         IN      A       191.5.157.33

;Liste des Types CNAME
www                             IN      CNAME   hqfwsrv.wsl2024.org
authentication                  IN      CNAME   hqdwsrv.wsl2024.org
```

Voici le fichier de la zone worldskills.org :

```
;Fichier config zone worldskills.org
;Tout ce qui suit un point virgule est un commentaire


$TTL 604800 ;Permet de maintenir le Time to live de notre zone a 604800 secondes, pendant 1 semaine,
@       IN      SOA     ns.worldskills.org. admin.worldskills.org. (
                        2               ;Serie
                        604800          ;TTL / Tmps de refresh data
                        86400           ;Retry
                        2419200         ;Expire
                        604800          ;Negative cache TTL

)

;Liste des Nameservers pour la zone
                                IN      NS      ns.worldskills.org.
ns                              IN      A       8.8.1.3

;Liste des Types A
inetsrv                         IN      A       8.8.1.1
wanrtr                          IN      A       8.8.1.6

;Liste des types CNAME
www                             IN      CNAME   inetsrv.worldskills.org
ftp                             IN      CNAME   inetsrv.worldskills.org
```

3. Zabbix

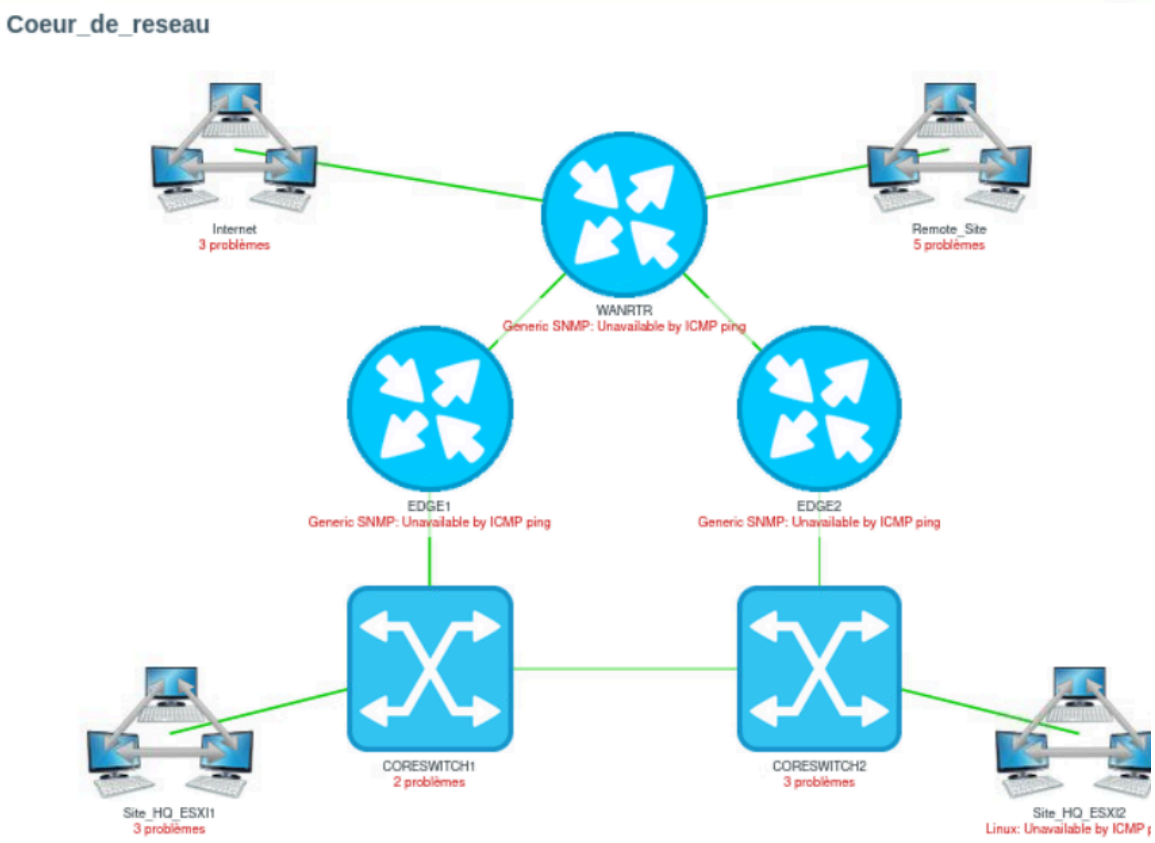
Zabbix est un logiciel de surveillance réseau open source. Il est conçu pour surveiller et suivre l'état des divers services réseau, serveurs et autres matériels réseau.

Dans cette Saé nous avons utilisé le protocole SNMP. Nous avons configuré sur tous nos serveurs, routeurs et switchs la même communauté SNMP “4P”.

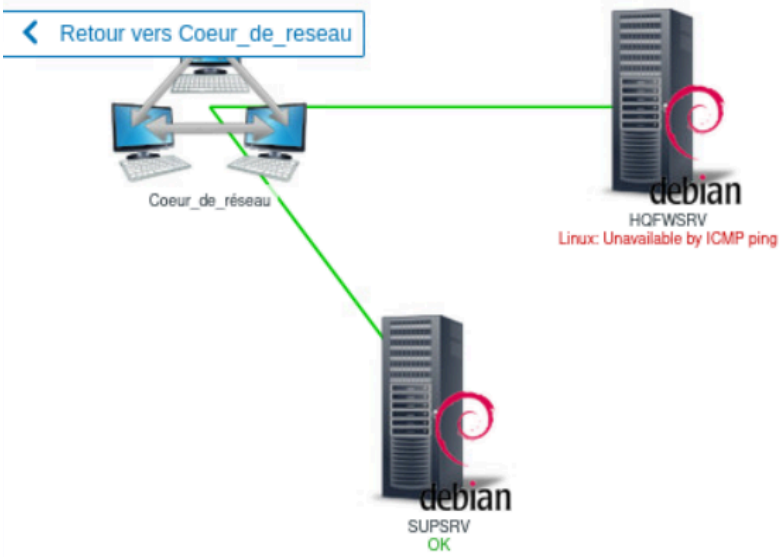
Le serveur Zabbix est installé sur le serveur SUPSRV dans le VLAN 99 qui est le VLAN de management. Il a donc accès à l'entièreté du réseau. Ainsi, il lui est possible de récupérer des données telles que l’utilisation du CPU, de la mémoire, l'utilisation des interfaces réseaux, etc.

CORESWITCH1	Éléments 341	Déclencheurs 155	Graphiques 36	Découverte 2	Web	10.1.99.12:161	Network Generic Device by SNMP	Activé
CORESWITCH2	Éléments 341	Déclencheurs 155	Graphiques 36	Découverte 2	Web	10.1.99.13:161	Network Generic Device by SNMP	Activé
DNSSRV	Éléments 32	Déclencheurs 10	Graphiques 5	Découverte 5	Web	8.8.1.3:161	Linux by SNMP	Activé
EDGE1	Éléments 113	Déclencheurs 52	Graphiques 11	Découverte 2	Web	10.1.254.245:161	Network Generic Device by SNMP	Activé
EDGE2	Éléments 221	Déclencheurs 100	Graphiques 22	Découverte 2	Web	10.1.254.249:161	Network Generic Device by SNMP	Activé
HQDCSRV	Éléments 69	Déclencheurs 32	Graphiques 9	Découverte 3	Web	10.1.10.1:161	Windows by SNMP	Activé
HQFWSRV	Éléments 32	Déclencheurs 10	Graphiques 5	Découverte 5	Web	10.1.10.3:161	Linux by SNMP	Activé
HQINFRASRV	Éléments 32	Déclencheurs 10	Graphiques 5	Découverte 5	Web	10.1.10.2:161	Linux by SNMP	Activé
HQMAILSRV	Éléments 32	Déclencheurs 10	Graphiques 5	Découverte 5	Web	10.1.10.4:161	Linux by SNMP	Activé
INETSRV1	Éléments 32	Déclencheurs 10	Graphiques 5	Découverte 5	Web	8.8.1.12:161	Linux by SNMP	Activé

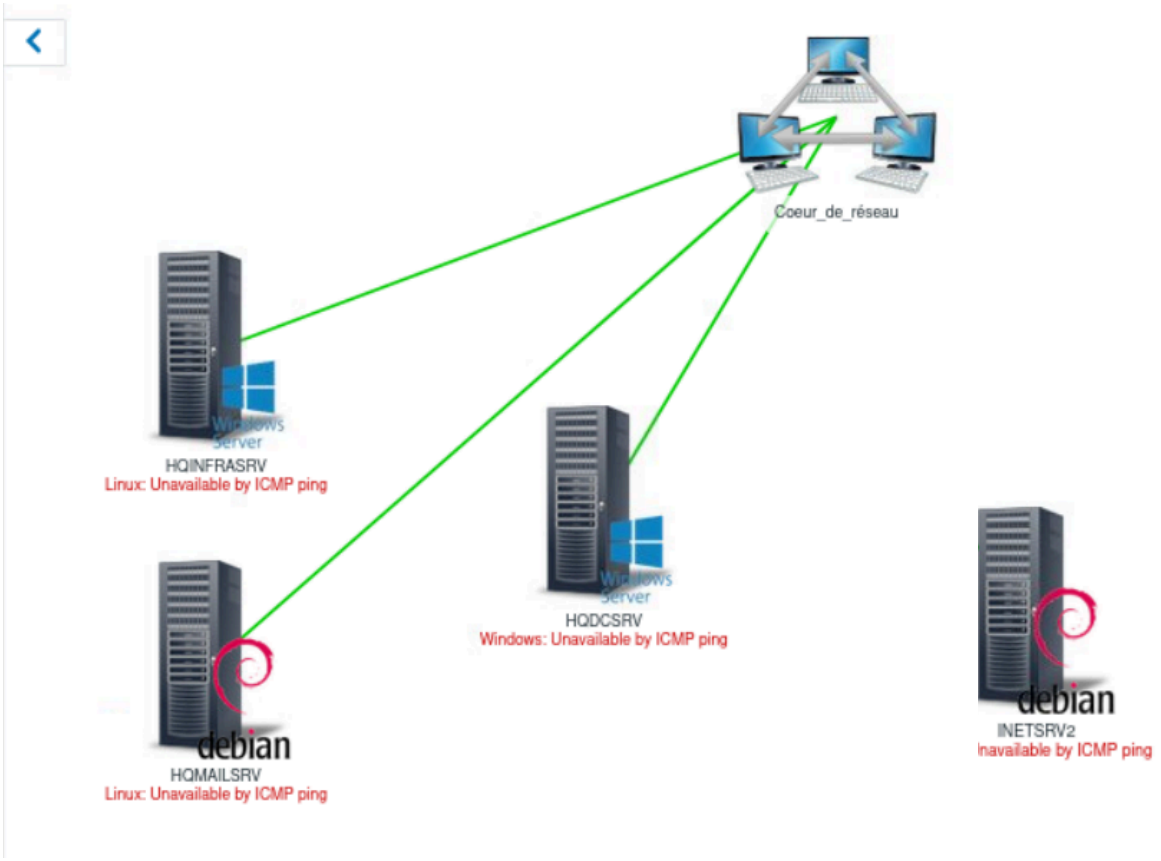
Après avoir récolté des données sur les équipements et serveurs du réseau, nous avons réalisé une cartographie dynamique du réseau.



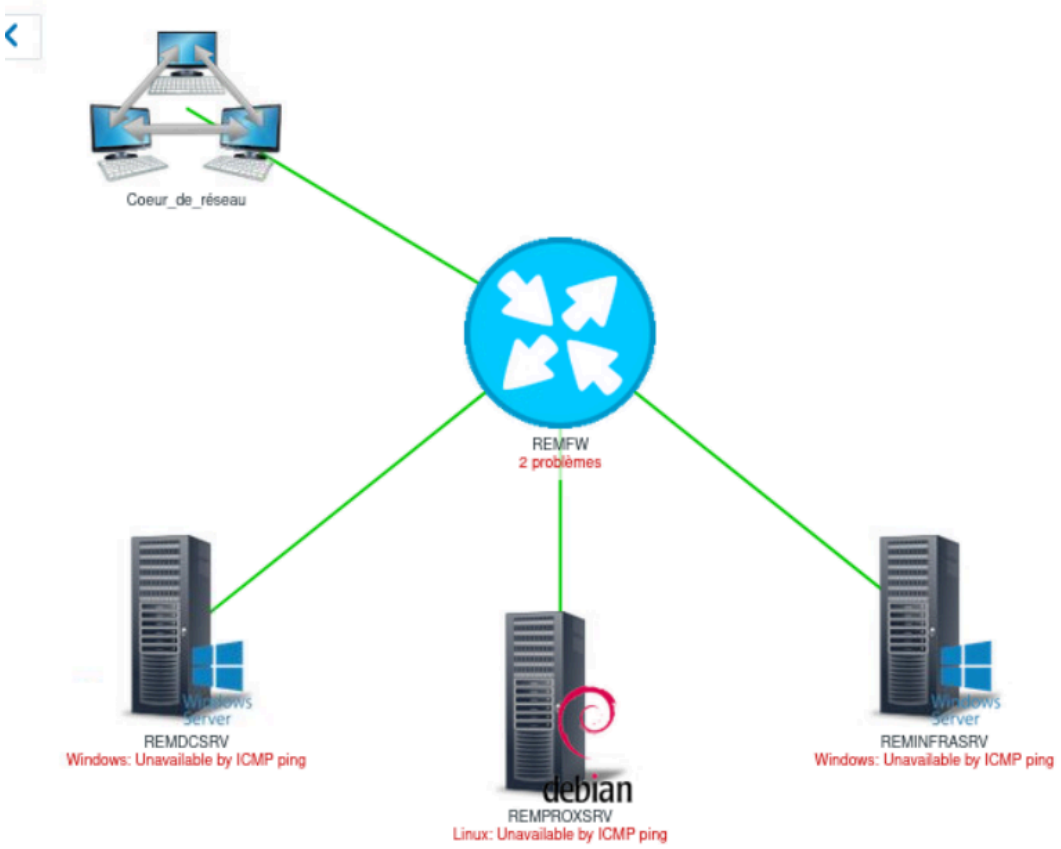
Cartographie du coeur de réseau



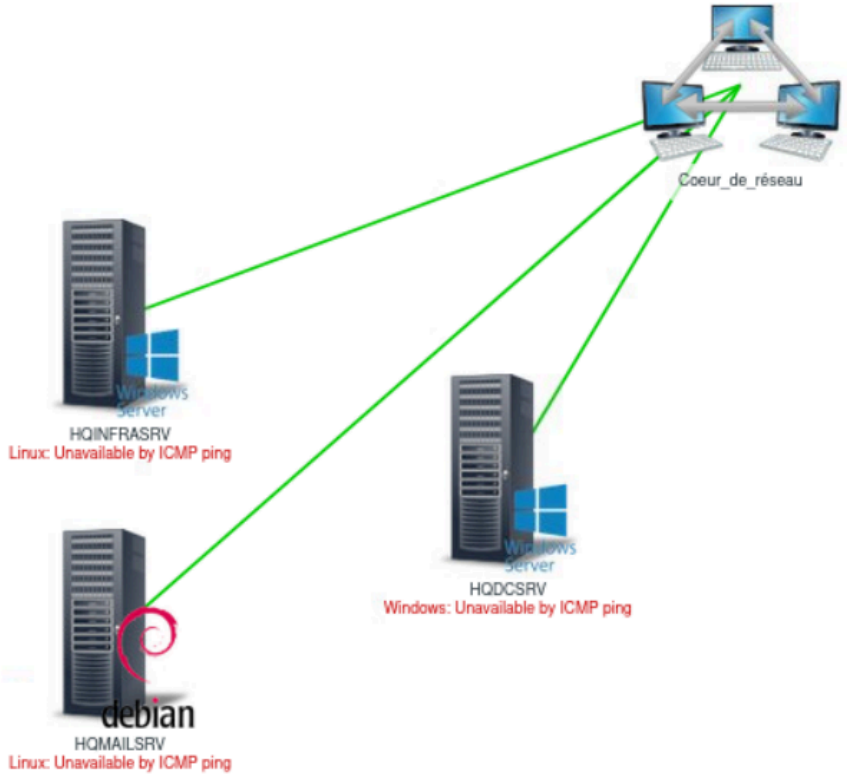
Cartographie du site HQ ESXI 2



Cartographie du site Internet

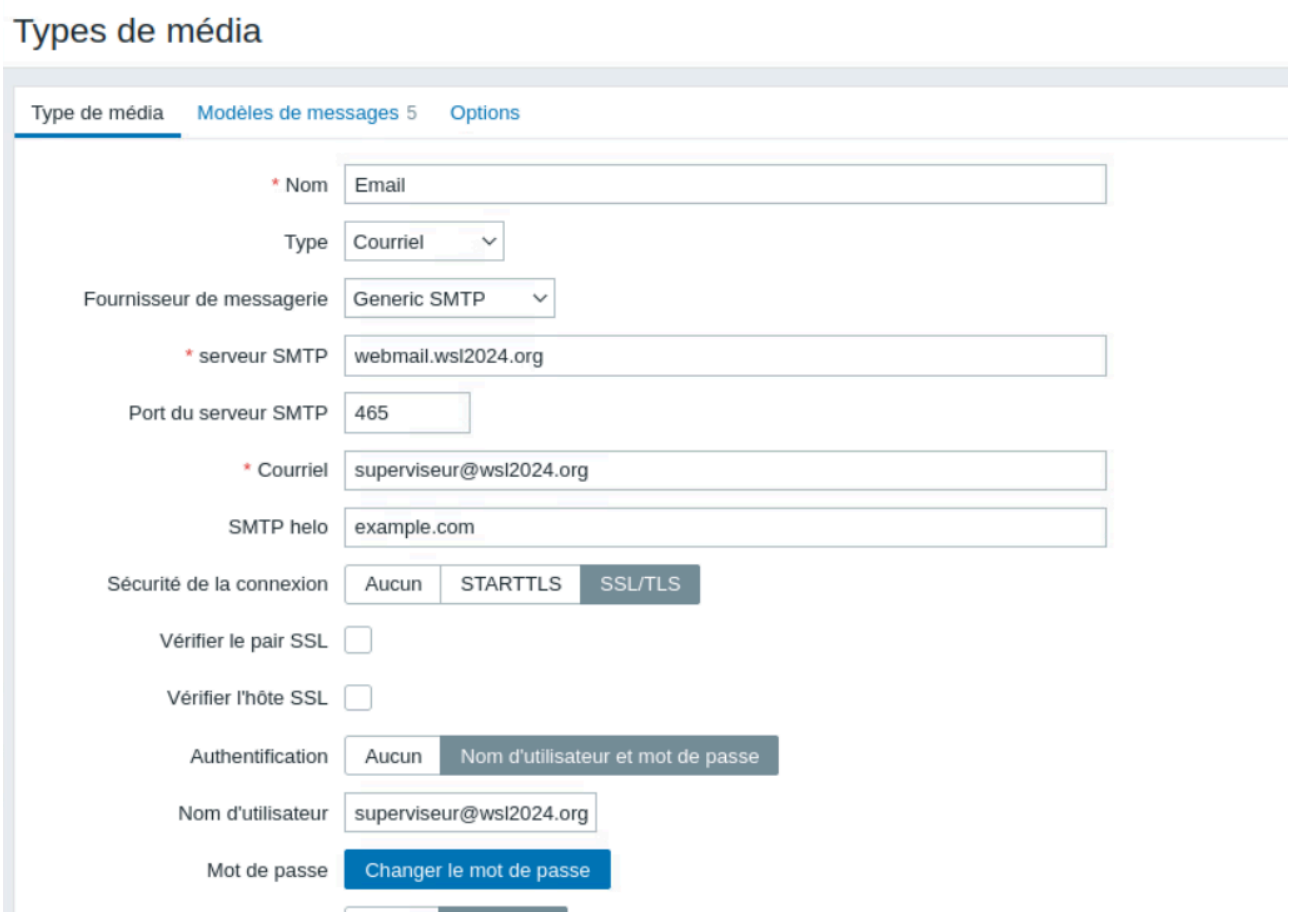


Cartographie du site Remote

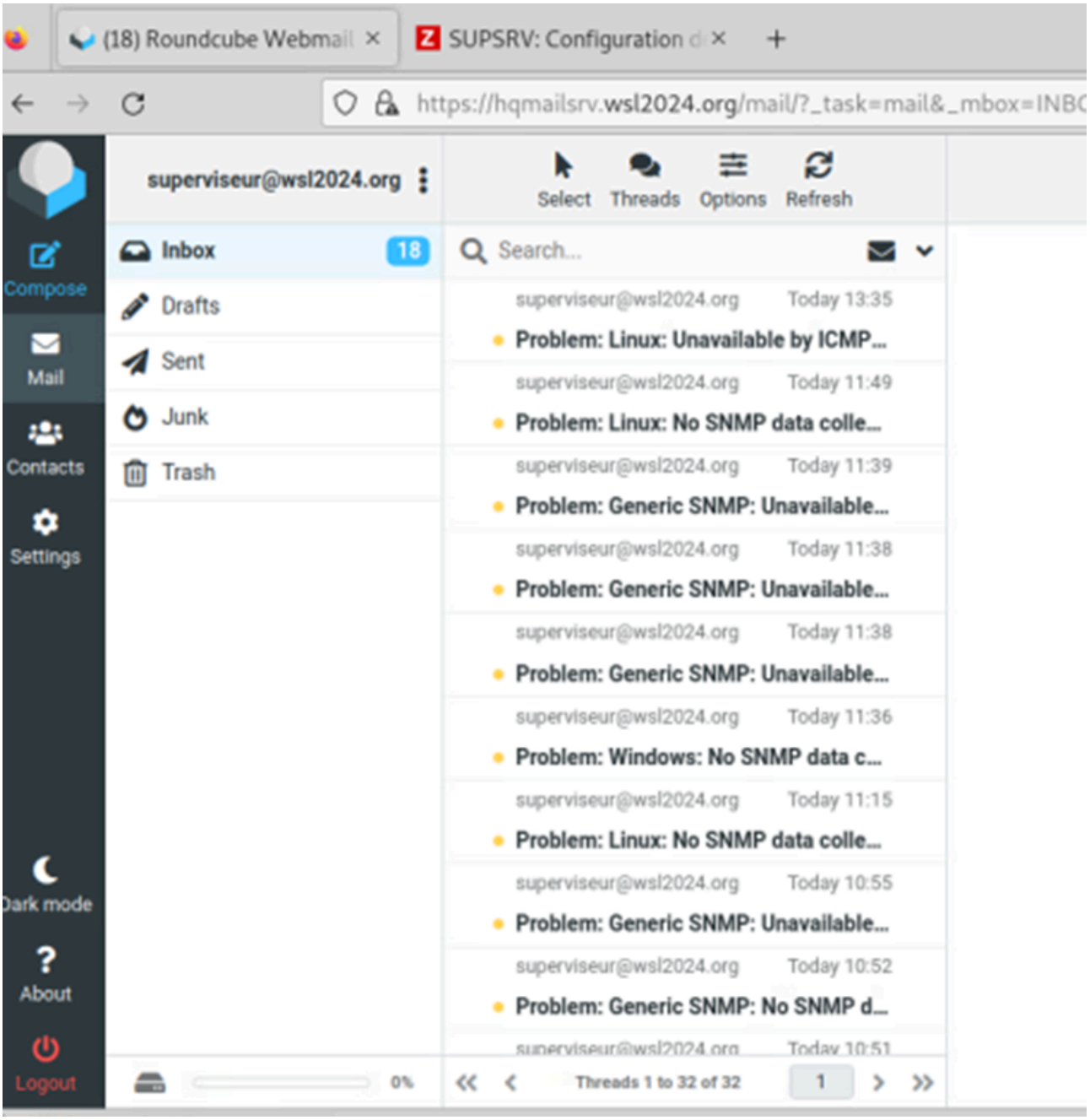


Cartographie du site HQ ESXI 1

Après avoir configuré la récupération des données, nous avons configuré l'envoi de mail par Zabbix à chaque incident sur le réseau à l'aide de SMTP.



Configuration de l'accès à la boîte mail de Zabbix.

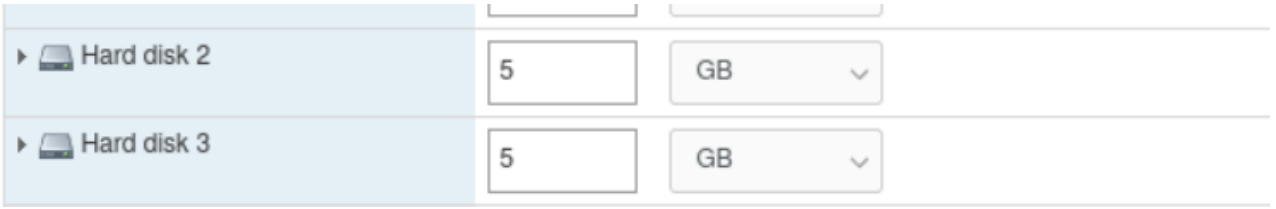


Résultat des mails d'alertes.

4. Samba

Nous avons ensuite mis en place un serveur de partage de fichier sur HQINFRASRV.
Pour ce faire nous avons utilisé la solution Samba.

Nous avons tout d'abord installé 2 nouveaux disques de 5 Go dans la machine.



Nous avons ensuite créé un groupe de disques nommé vgstorage dans lequel nous avons inclus les deux disques nouvellement créés.

Par la suite, nous avons créé deux volumes logique de 2 Go chacun :

- /dev/vgstorage/lvdatastorage
- /dev/vgstorage/lviscsi

Nous avons ensuite configuré le format de lvdatastorage pour qu'il soit en ext4 et nous avons terminé par monter le volume lvdatastorage.

Nous avons ensuite créé 2 dossiers, Public et Private

```
root@HQINFRASRV:~# lvdisplay
  --- Logical volume ---
  LV Path                /dev/vgstorage/lvdatastorage
  LV Name                lvdatastorage
  VG Name                vgstorage
  LV UUID                GQC0Re-66jd-5krU-Mv7A-UgQp-uDsU-FMoTyL
  LV Write Access        read/write
  LV Creation host, time HQINFRASRV, 2023-12-14 14:02:16 +0100
  LV Status              available
  # open                 0
  LV Size                2,00 GiB
  Current LE             512
  Segments               1
  Allocation             inherit
  Read ahead sectors     auto
  - currently set to     256
  Block device           254:0

  --- Logical volume ---
  LV Path                /dev/vgstorage/lviscsi
  LV Name                lviscsi
  VG Name                vgstorage
  LV UUID                8vZ1fb-ofUb-nzdn-g84w-O8rz-ZphB-TwmpZl
  LV Write Access        read/write
  LV Creation host, time HQINFRASRV, 2023-12-14 14:03:39 +0100
  LV Status              available
  # open                 0
  LV Size                2,00 GiB
  Current LE             512
  Segments               1
  Allocation             inherit
  Read ahead sectors     auto
  - currently set to     256
  Block device           254:1
```

Nous avons rencontré un problème par la suite, plus spécifiquement lors de la démonstration.
Lors de cette dernière, nous nous sommes rendu compte que le serveur samba n'était pas démarré, on s'est rendu compte par la suite que nous avions oublié d'ajouter le disque partagé dans fstab ce qui lui permet de monter automatiquement au démarrage du serveur.

Ce problème peut être réglé en ajoutant la ligne suivante dans le fichier /etc/fstab :

```
UUID=6cba3c7d-b628-4129-97f2-b34b66fe5701       none    ext4    defaults
```

La ligne est partagée en 4 parties :

- La première sert à indiquer l'uuid du disque
- La deuxième sert à
- La troisième sert à définir le format, dans notre cas EXT4
- La quatrième sert à définir le montage du disque, en sélectionnant "defaults" cela fait en sorte que le disque se monte automatiquement au démarrage de la machine .

Nous avons ensuite créé plusieurs utilisateurs à qui on a par la suite, chacun de ses utilisateurs possède le même mot de passe : P@ssx0rd

```
Jean:x:1000:1000::/home/Jean:/bin/sh
Tom:x:1001:1001::/home/Tom:/bin/sh
Emma:x:1002:1002::/home/Emma:/bin/sh
```

Nous avons ensuite attribué les droits sur les fichiers en éditant le fichier /ect/samba/smb.conf

```
[Public]
        path = /srv/datastorage/shares/public
        read only = yes

[Private]
        path = /srv/datastorage/shares/private
        browable = no
        valid users = Tom Emma
        read list = Jean
        write list = Tom Emma
        veto files = /.exe/.zip
```

5. Asterisk

Nous avons ensuite voulu installer un serveur de téléphonie avec la solution Asterisk.
Pour ce faire, nous avons commencé par réaliser un plan de téléphonie avec les numéros de chaque utilisateurs

Prénon NOM	Service	Site	N°	N° Service
Vincent TIM	IT	HQ	101	301
Ness PRESSO	Direction	HQ	102	302
Jean TICIPE	Factory	HQ	103	303
Rick OLA	Sales	HQ	104	304
Ela STIQUE	Warehouse	REM	201	305
Clotilde Morir	Direction	REM	202	302
Denis Peltier	IT	REM	203	301

```
[default](!)
type=friend
;password=azerty
host=dynamic
qualify=no
secret=P@ssw0rd
nat=no
canreinvite=yes

[101](default)
fullname=Vincent TIM
username=vtim
context=IT

[102](default)
fullname=Ness PRESSO
username=npresso
context=Direction

[103](default)
fullname=Jean TICIPE
username=jticipe
context=Factory

[104](default)
fullname=Rick OLA
username=rola
context=Sales
```

Une fois cela fait, nous avons édité le fichier sip.conf avec la configuration adaptée au cas par cas.

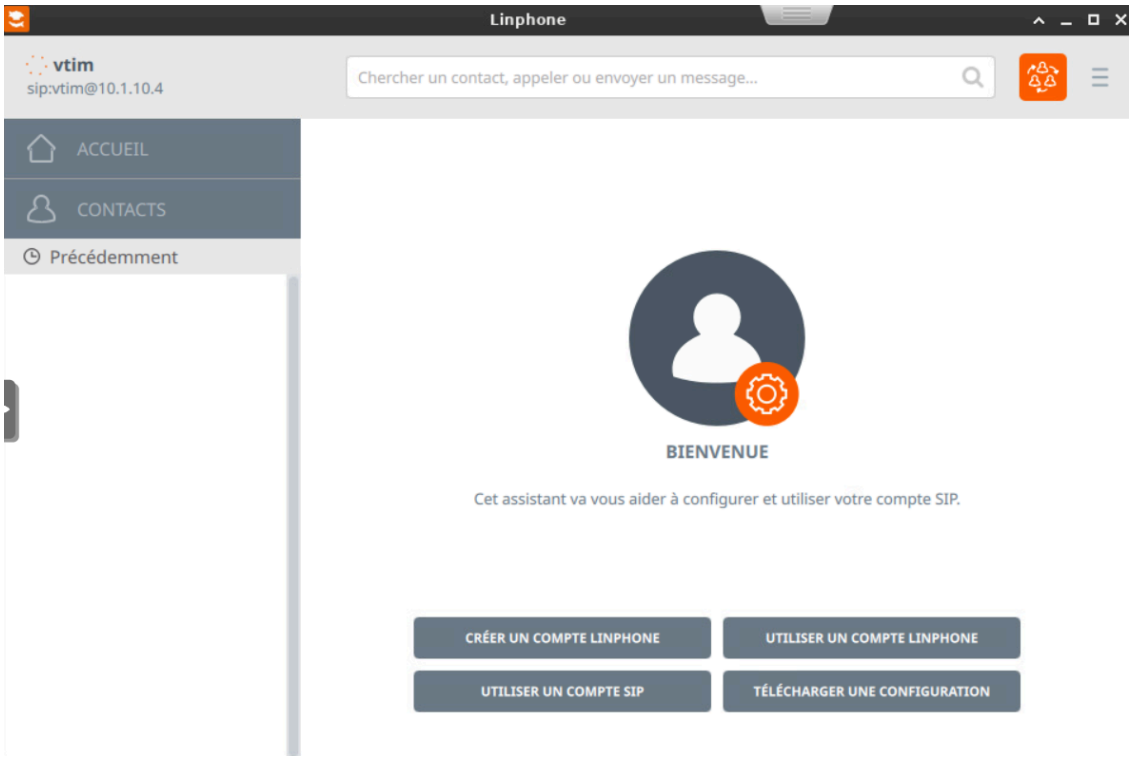
Nous avons commencé par définir un template “default” dans lequel nous avons renseigné toutes les configurations communes à tous les utilisateurs.

Puis nous avons configuré les différents utilisateurs.

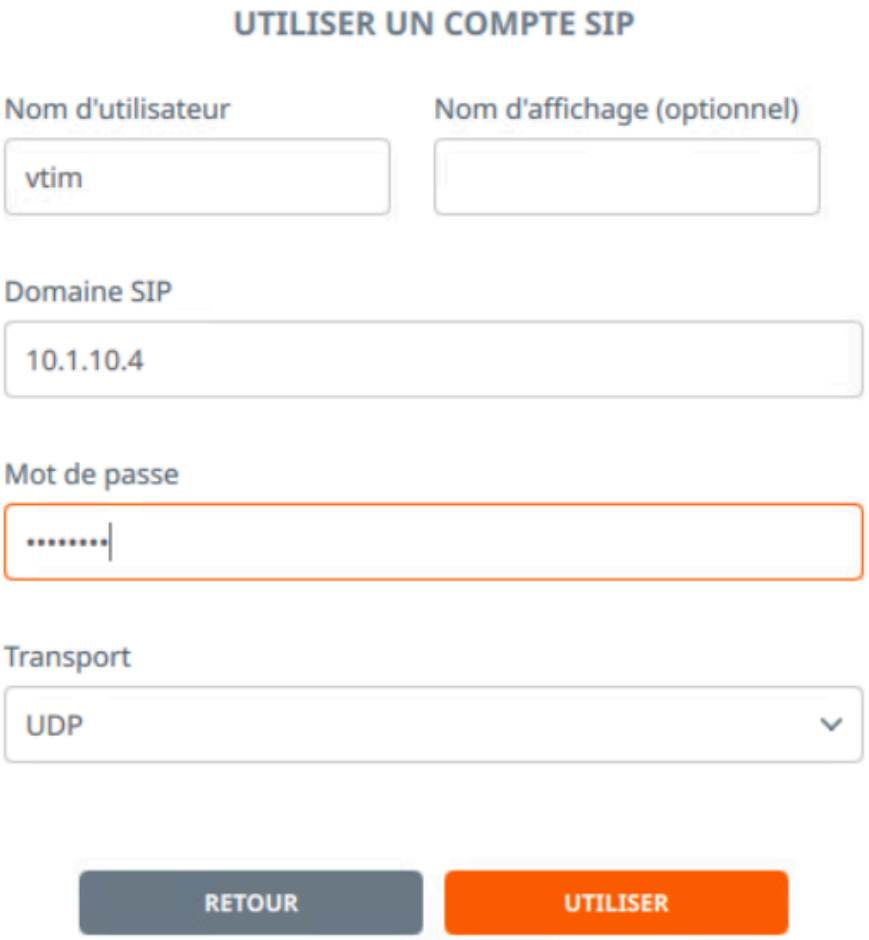
Suite à cela nous redémarré le dialplan avec les commandes suivantes :

- `sip reload`
- `dialplan reload`

Nous avons ensuite télécharge les softphones de la solution Linphone sur les machines suivantes : HQINFRASRV et HQCLT



Nous avons ensuite tenté de nous connecter avec un compte défini précédemment.



Mais malheureusement ça n'a pas fonctionné.

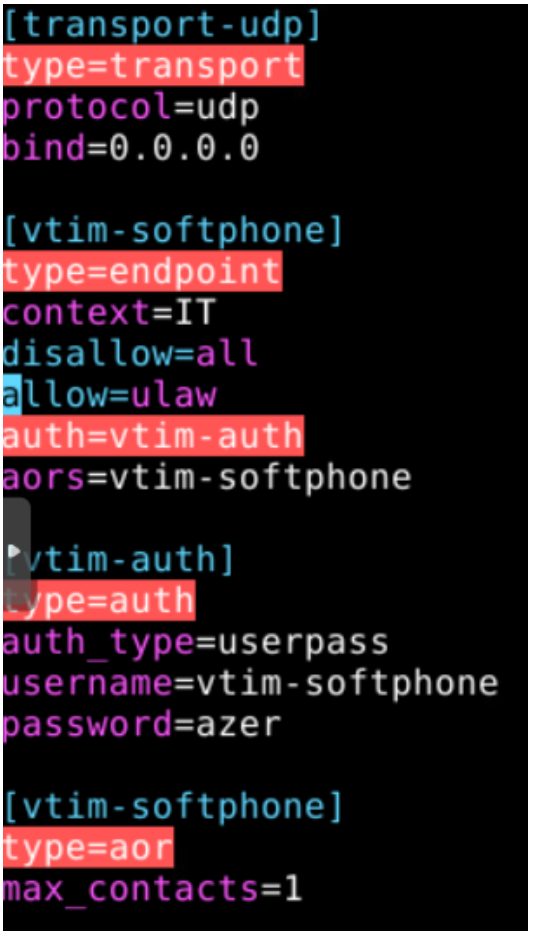
Pour remédier à ce problème nous avons essayé de réaliser une deuxième une deuxième configuration mais cette fois avec pjsip qui est la version la plus récente de SIP.

Après avoir regardé plusieurs documentation, nous sommes arrivés au résultat suivant.

Après cela nous avons redémarrer le service avec les commandes suivantes :

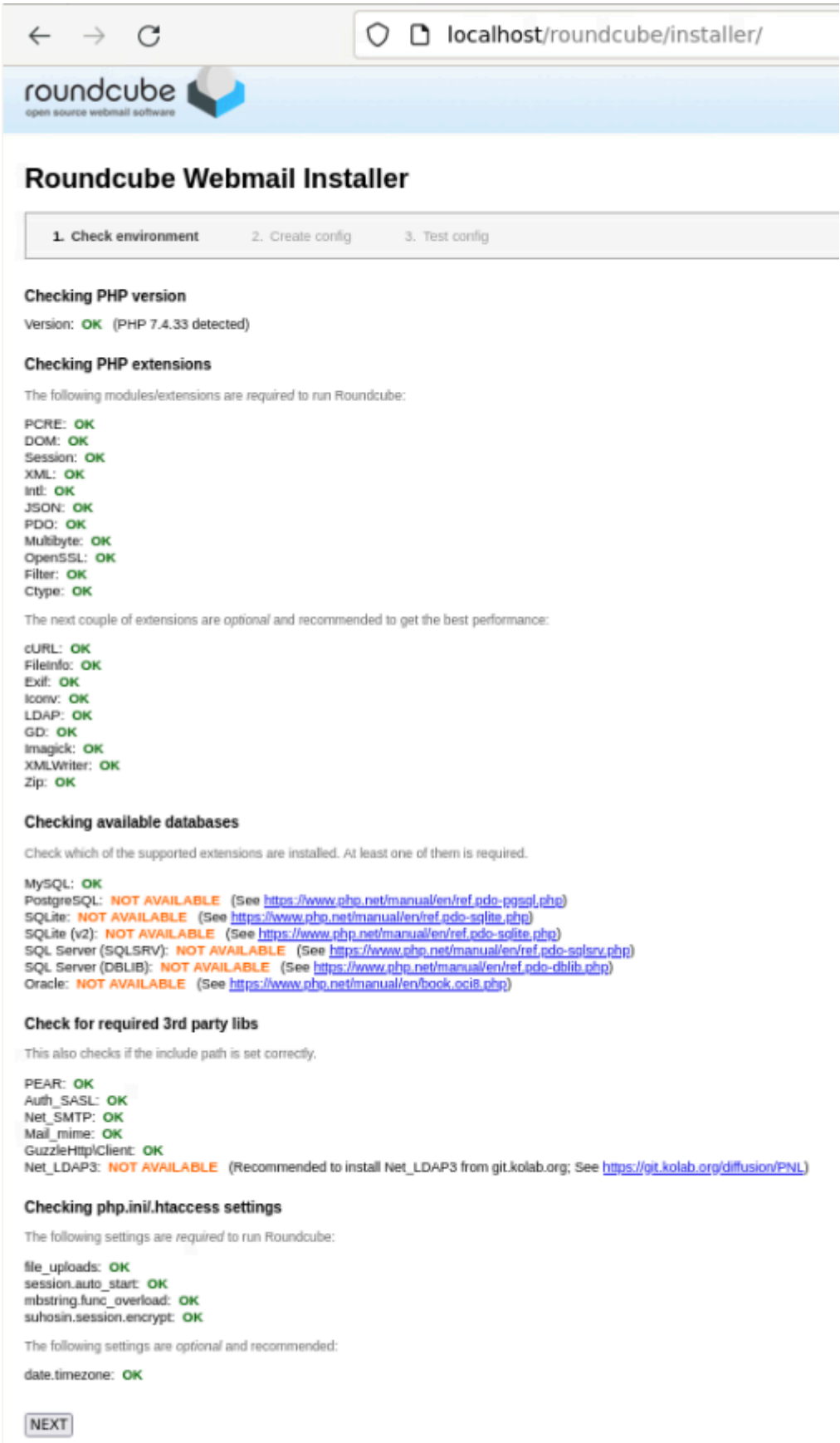
- `dialplan reload`
- `pjsip reload`

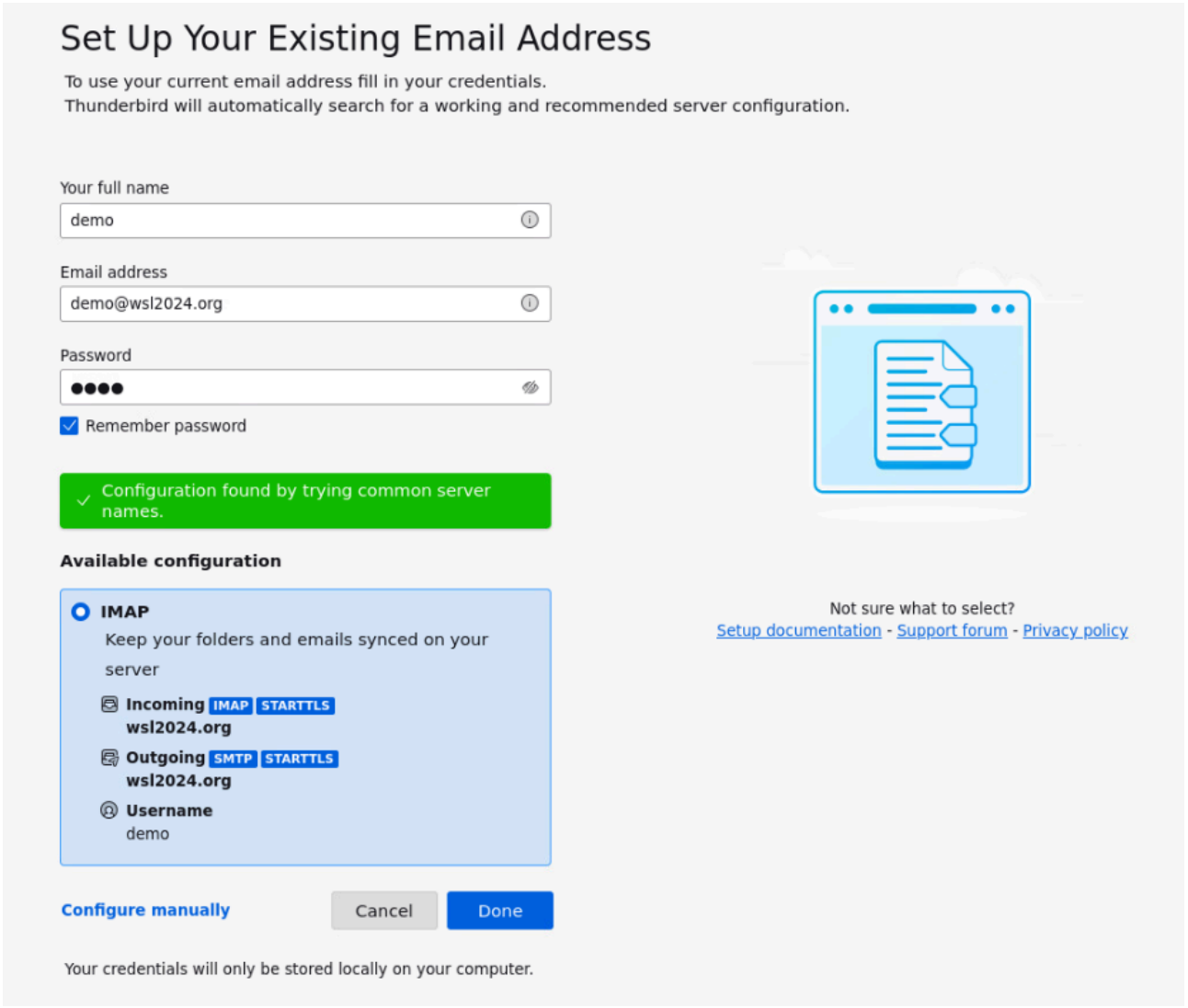
Nous avons ensuite essayé de reconnecter l'utilisateur sur Linphone mais nous avons obtenu exactement le même résultat



Nous n'avons malheureusement pas réussi à faire fonctionner ce service avant la fin de la DAE

6. WEBMAIL


localhost/roundcube/installer/
roundcube
open source webmail software
Roundcube Webmail Installer
1. Check environment
2. Create config
3. Test config
Checking PHP version
Version: OK (PHP 7.4.33 detected)
Checking PHP extensions
The following modules/extensions are required to run Roundcube:
PCRE: OK
DOM: OK
Session: OK
XML: OK
Intl: OK
JSON: OK
PDO: OK
Multibyte: OK
OpenSSL: OK
Filter: OK
Ctype: OK
The next couple of extensions are optional and recommended to get the best performance:
cURL: OK
FileInfo: OK
Exif: OK
Iconv: OK
LDAP: OK
GD: OK
Imagick: OK
XMLWriter: OK
Zip: OK
Checking available databases
Check which of the supported extensions are installed. At least one of them is required.
MySQL: OK
PostgreSQL: NOT AVAILABLE (See https://www.php.net/manual/en/ref.pdo-pgsql.php)
SQLite: NOT AVAILABLE (See https://www.php.net/manual/en/ref.pdo-sqlite.php)
SQLite (v2): NOT AVAILABLE (See https://www.php.net/manual/en/ref.pdo-sqlite.php)
SQL Server (SQLSRV): NOT AVAILABLE (See https://www.php.net/manual/en/ref.pdo-sqlsrv.php)
SQL Server (DBLIB): NOT AVAILABLE (See https://www.php.net/manual/en/ref.pdo-dblib.php)
Oracle: NOT AVAILABLE (See https://www.php.net/manual/en/book.oci8.php)
Check for required 3rd party libs
This also checks if the include path is set correctly.
PEAR: OK
Auth_SASL: OK
Net_SMTP: OK
Mail_mime: OK
GuzzleHttp\Client: OK
Net_LDAP3: NOT AVAILABLE (Recommended to install Net_LDAP3 from git.kolab.org; See https://git.kolab.org/diffusion/PNL)
Checking php.ini/.htaccess settings
The following settings are required to run Roundcube:
file_uploads: OK
session.auto_start: OK
mbstring.func_overload: OK
suhosin.session.encrypt: OK
The following settings are optional and recommended:
date.timezone: OK
NEXT


Set Up Your Existing Email Address
To use your current email address fill in your credentials.
Thunderbird will automatically search for a working and recommended server configuration.
Your full name
demo
Email address
demo@wsl2024.org
Password
Remember password
Configuration found by trying common server names.
Available configuration
IMAP
Keep your folders and emails synced on your server
Incoming IMAP STARTTLS
wsl2024.org
Outgoing SMTP STARTTLS
wsl2024.org
Username
demo
Configure manually
Cancel
Done
Your credentials will only be stored locally on your computer.
Not sure what to select?
Setup documentation - Support forum - Privacy policy

7. FTP

Le serveur FTP a été configuré avec le service *proftpd* après une sauvegarde du fichier de configuration de base.
La configuration FTP permet à l'utilisateur devops de se connecter en ayant les droits d'écriture et de lecture.
L'utilisateur devops est chroot dans dans son répertoire home.

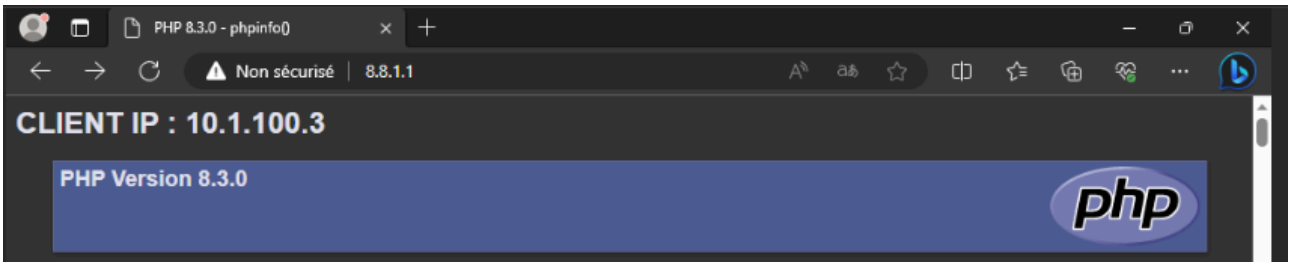
```
ServerName "ftp.worldskills.org"
DefaultAddress 0.0.0.0
ServerType standalone
DefaultServer on
RequireValidShell off
AuthPAM off
AuthPAMConfig devops
Port 21
Umask 022
MaxInstances 30
User devops
Group devops
DefaultRoot ~

AllowOverwrite on
<IfModule mod_tls.c>
        TLSEngine on
        TLSLog /var/log/proftpd/tls/log
        TLSProtocol TLSv1 TLSv1.1 TLSv1.2
        TLSVerifyClient off

        TLSRequired off
        TLSRSACertificateFile /etc/proftpd/FTP/cert/ftp.crt
        TLSRSACertificateKeyFile /etc/proftpd/FTP/private/ftp.key
        TLSPassPhraseProvider /etc/proftpd/FTP/pass
</IfModule>

<Limit SITE_CHMOD>
        DenyALL
</Limit>

<Anonymous ~devops>
        User devops
        Group devops
        UserAlias anonymous devops
        MaxClients 10
        <Limit WRITE>
                AllowUser devops
                DenyALL
        </Limit>
</Anonymous>
```

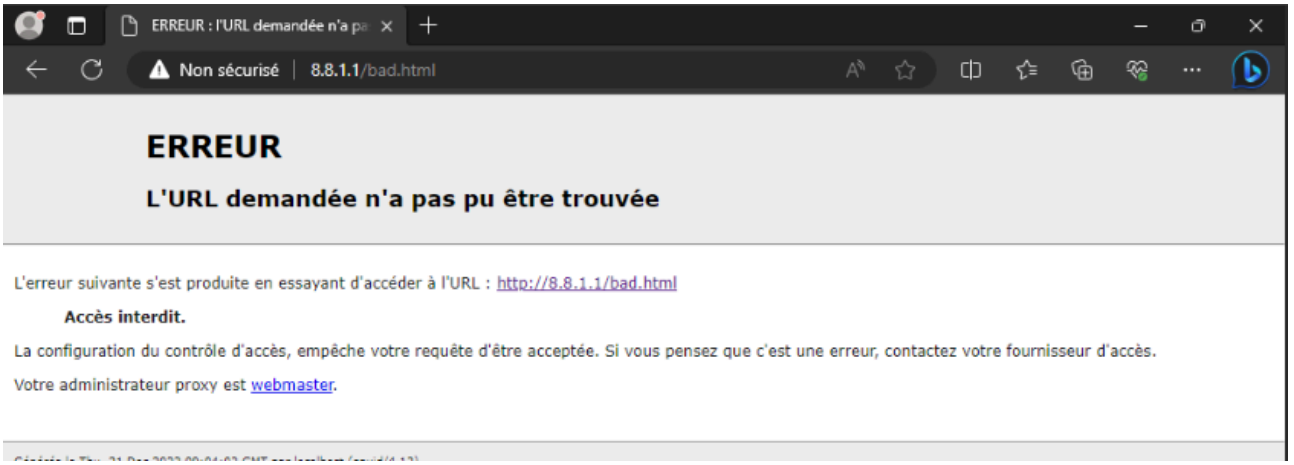
8. Proxy Squid

Le proxy squid permet de faire des access list (ACL) sur les sources/destinations sur la base d'adresse IP et de noms de domaine et aussi peut filtrer avec du regex les url.

```
http_access allow localhost
acl wh dstdomain .worldskills.org .wsl2024.org
http_access allow wh


############################################################
acl block_bad url_regex "/etc/squid/bad.txt"
http_access deny block_bad
http_access allow localhost
acl mynet dst 8.8.0.0/16
http_access allow mynet

############################################################
```

```
http_port 3128
```





c. Sécurité

i. Pare-feux

1. HQFWSRV

Pour configurer HQFWSRV, nous avons installé le paquet IPtable.
La configuration du service passait par le création d'un script Bash, celui-ci se décompose en 2 parties.

```
#Setup Prerouting
$it -t nat -A PREROUTING -i ens224 -p tcp -d 217.1.160.1 --dport 80 -j DNAT --to 10.1.30.1:80;
$it -t nat -A PREROUTING -i ens224 -p tcp -d 217.1.160.1 --dport 443 -j DNAT --to 10.1.30.1:443;

#Setup Postrouting
$it -t nat -A POSTROUTING -o ens224 -j MASQUERADE;
$it -t nat -A POSTROUTING -s 10.1.30.1/30 ! -d 10.1.30.1/30 -j MASQUERADE;
```

- **PREROUTING :**

Le prerouting consiste en un changement d'adresse IP. Lorsqu'un paquet arrive sur le pare-feux, l'adresse de destination est remplacée par une adresse d'un réseau pour pouvoir par exemple avoir accès à un service dans une DMZ isolé.

- **POSTROUTING :**

Le postrouting consiste à translater les adresses sources en une autre, typiquement ce que fait une passerelle internet en changeant l'adresse du client dans un réseau local avec l'adresse publique de la passerelle.

Nous avons également installé ufw pour compléter la configuration IPtable. Nous avons défini deux règles autorisant le trafic TCP de n'importe quelle source vers nos serveurs web (adresse HSRP) sur les ports 80 et 443.

```
#!/bin/bash

ufw allow proto tcp from any to 217.1.160.1 port 80;
ufw allow proto tcp from any to 217.1.160.1 port 443;
```

d. Haute Disponibilité

i. HSRP et VRRP

Lorsque l'on souhaite avoir une certaine qualité de service sur son réseau, il est nécessaire de mettre en place des mécanismes comme de l'équilibrage de charge ou de la redondance. C'est pour cette raison que nous avons configuré le protocole HSRP permettant à plusieurs équipements de partager la même adresse IP virtuelle. De ce fait, lorsque l'équipement principal arrête de fonctionner un des autres équipements peut prendre le relais de manière transparente pour les utilisateurs.

Nous avons configuré cette fonctionnalité sur plusieurs équipements de notre réseau.

Tout d'abord entre nos deux CoreSwitch sur leurs interfaces VLAN.


```
interface Vlan10
 ip address 10.1.10.60 255.255.255.192
 standby 10 ip 10.1.10.62
 standby 10 priority 110
 standby 10 preempt
!
interface Vlan20
 ip address 10.1.21.252 255.255.254.0
 ip helper-address 10.1.10.2
 ip helper-address 10.1.20.1
 standby 20 ip 10.1.21.254
 standby 20 priority 110
 standby 20 preempt
```

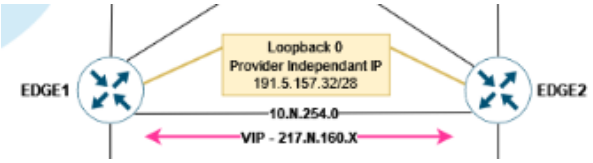
Pour configurer une adresse IP HSRP, on utilise les commandes standby. Pour chaque IP partagée, on doit préciser le numéro de session HSRP.

On doit préciser l'adresse IP partagée.

On doit également configurer une priorité, plus le chiffre est élevé plus l'équipement est prioritaire.

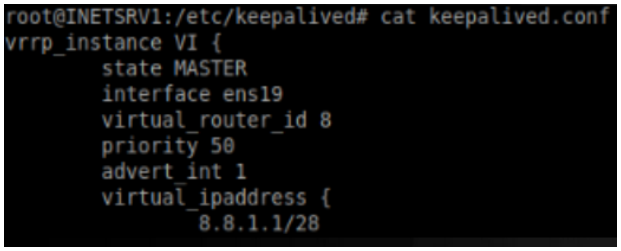
Finalement, le paramètre standby permet de faire reprendre le trafic à l'équipement principal lorsqu'il revient sur le réseau.

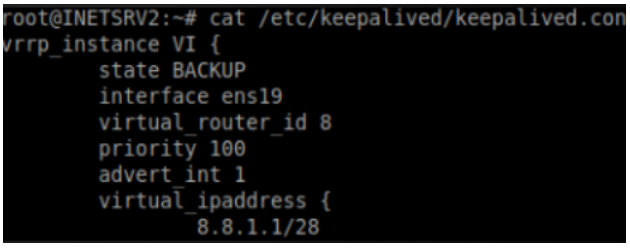
La même configuration a été réalisée sur le switch pair. Une configuration similaire a également été faite sur nos routeurs, celle-ci est utilisée comme passerelle pour le routage inter-VRF.



Pour permettre d'utiliser nos serveurs web de manière transparente en cas de panne, nous avons configuré de la haute disponibilité sur celui-ci en mettant en place une adresse partagée via le protocole VRRP grâce au paquet linux keepalived.

Le fonctionnement est similaire à celui du HSRP Cisco, la seule différence étant la priorité des équipements. En effet, l'équipement avec la valeur la plus basse est prioritaire.





L'équipement actif doit également être précisé avec le champ state (Master pour principal, Backup pour secondaire). Nous avons donc déclaré le nom DNS du serveur web sur l'adresse IP HSRP.

ii. Docker

Le paradigme de docker est basé sur les services donc une image ou container représente un service qui est défini par un Dockerfile.

Docker Compose rassemble un groupe de containers qui interagissent ensemble.
Pour pouvoir utiliser docker derrière un proxy il faut renseigner dans le daemon docker le proxy

/etc/systemd/system/docker.service.d/http-proxy.conf
[Service] Environment="HTTP_PROXY=http://194.57.85.1:3128" Environment="HTTPS_PROXY=http://194.57.85.1:3128"

puis faire `systemctl daemon-reload` puis `systemctl restart docker`

Dans le répertoire de chaque utilisateur il faut encore renseigner le proxy pour utiliser docker

.docker/config.conf (home directory of linux users)

{
"proxies": {
"default": {
"httpProxy": "http://webcache.pu-pm.univ-fcomte.fr:3128",
"httpsProxy": "https://webcache.pu-pm.univ-fcomte.fr:3128",
"noProxy": ""
}
}
}

Pour pouvoir utiliser cette image par la suite il faut faire la commande :

docker build -t nginx Dockerfile

FROM nginx:latest COPY nginx/default.conf /etc/nginx/conf.d/ COPY nginx/INETSRV1.pem /etc/ssl/certs/ COPY nginx/INETSRV1.key /etc/ssl/private/ COPY nginx/pass /etc/ssl/pass EXPOSE 80 EXPOSE 443	Récupération de l'image NGINX copier depuis l'hôte la configuration NGINX dans le container Copier les certificats dans le container Copier le fichier contenant le mot de passe du certificat L'image expose les ports 80 et 443 sur la machine hôte

Pour lancer le docker compose il faut faire :

```
docker compose build
docker compose run -d
```

le *-d* c'est pour lancer les docker en mode daemon.

version: "v1.1"
 	
services:
 	
  web:
  build:
  context: .
 	
  dockerfile: nginx/Dockerfile
  ports:
  - '80:80'
  - '443:443'
  volumes:
  - /app:/var/www/html/
  networks:
  - internal
  php:
  image: php:8-fpm
  volumes:
  - /app:/var/www/html/
  networks:
  - internal
networks:
  internal:
  driver: bridge
	Versionnage (format libre à l'utilisateur)
	
	définition des services (nécessaire)
	
	Nom d'un service mis en place
	définition de l'image utilisé pour le service web
	se place dans le répertoire du docker-compose.yml
	utilise l'image défini précédemment
	expose les ports pour ce service donc NGINX
	
	créer un répertoire dans la racine de la machine hôte lié à /var/www/html du container
	ajoute ce service dans le réseau internal
	
	Deuxième service donc une autre image prise de php-fpm
	
	définition du réseau internal en mode bridge

iii. Failover

Nous avons ensuite fait la configuration du DHCP failover. Celui-ci avait deux buts notables :

- Partager la plage d'adresse IP en deux parties distinctes
- Avoir un serveur DHCP de backup

Pour ce faire, nous avons modifier la configuration du serveur DHCP de HQINFRASRV pour la pool du failover ainsi qu'inclure la configuration du failover que je vais détailler après.

```
include "/etc/dhcp/failover.conf";

default-lease-time 600;
max-lease-time 7200;

subnet 10.1.20.0 netmask 255.255.254.0 {
#       range 10.1.20.207 10.1.21.254;
        pool {
                range 10.1.20.207 10.1.21.101;
                failover peer "dhcp-failover";
        }
        option routers 10.1.21.254;
        option domain-name-servers hqdcsrv.hq.wsl2024.org;
        option domain-name "hq.wsl2024.org";
        option ntp-servers hqinfrasrv.wsl2024.org;
        default-lease-time 7200;
        max-lease-time 7200;
}
```

Dans la configuration du failover nous avons rajouté pour le serveur principal :

- Primary pour dire que ce serveur est le primaire
- L'adresse ip du serveur primaire
- Le port du DHCP
- Peer adresse pour définir l'adresse ip du deuxième serveur
- Peer port pour le port du DHCP du deuxième serveur
- Le délai maximum de réponse
- La temporisation jusqu'à ce que les serveurs DHCP soit basculés
- Le mclt pour minimum cache lifetime interval
- Split pour dire où est ce qu'on coupe la plage IP

```
failover peer "dhcp-failover" {
        primary;
        address 10.1.10.2;
        port 647;
        peer address 10.1.10.4;
        peer port 647;
        max-response-delay 60;
        max-unacked-updates 10;
        mclt 1800;
        split 250;
}
```

Mais cela n'a pas fonctionné et nous avons obtenu une erreur nous disant qu'on se trouvait dans le mauvais subnet .
Nous n'avons malheureusement pas réussi à résoudre ce problème avant la fin du temps imparti.

e. Ansible

Pour automatiser des actions sur nos serveurs linux ou nos routeurs nous avons développé des "scripts" Ansible. Nous avons écrit des instructions simples appelées "playbooks" en YAML, un format de données facile à lire et à écrire. Ces playbooks définissent les tâches à effectuer et sont exécutables par Ansible. Lors de l'exécution du playbook il faut également exécuter un fichier .ini (inventaire) contenant les variables liées aux équipements comme l'os utilisé, l'ip de l'équipement, etc.

```
[cisco_routers]
EDGE1 ansible_host=10.1.254.253 ansible_user=ansible ansible_password=automation ansible_network_os=ios
WANRTR ansible_host=10.1.254.254 ansible_user=ansible ansible_password=automation ansible_network_os=ios
```

```
# credentials.yml

ansible_user: ansible
ansible_password: automation
```

Nous avons également préparé un fichier credentials.yml qui contient les logins et mots de passe des appareils.

Lors de l'exécution de notre playbook Update_upgrade_apt.yml avec la commande `sudo ansible-playbook -i inventaire.ini Update_upgrade.yml`, le script va se connecter en ssh sur la machine linux à update/upgrade, il va ensuite exécuter les commandes sudo apt update puis sudo apt upgrade.

Lors de l'exécution de notre playbook Récupération_FTP.yml avec la commande `ansible-playbook -i ftp_inventory.ini Récupération_FTP.yml`, le script va se connecter en ssh sur les routeurs du

réseau, puis exécuter la commande sh run pour afficher la configuration actuelle puis il va récupérer cette configuration dans un fichier texte et ensuite établir une connexion ftp avec le serveur INITSRV1 pour lui transmettre la configuration des routeurs.

Pour finir nous avons configuré le fichier crontab-e avec les lignes

```
0 0 * * * ansible-playbook -i /home/tp/tftp/ftp_inventory.ini /home/tp/tftp/Récupération_FTP.yml
0 0 * * * sudo ansible-playbook -i /home/tp/Bureau/ansible/update/inventaire.ini Update_upgrade_apt.yml
```

Ces deux lignes permettent d'exécuter automatiquement les 2 playbooks Ansible à minuit.

6. Pistes d'amélioration

Nous avons plusieurs points à améliorer dans notre infrastructure.

Dans un premier temps, nous pouvons automatiser entièrement nos configurations et tests gérés par Ansible afin de faciliter le déploiement de nouveaux appareils.

Au cours de ce projet, nous aurions dû prioriser la partie démonstration du projet. Nous aurions donc pu limiter la production de nouveaux jalons pour miser sur un maximum de services fonctionnels.

Nous avons décidé de prioriser un maximum de fonctionnel et avons donc mis de côté la partie sécurité du réseau. Nous pourrions donc implémenter un maximum de sécurité sur notre réseau comme par exemple ajouter un maximum de certificats, sécurisé FTP avec FTPS, implémenter ADCS.

Nous aurions pu réaliser un audit de sécurité de notre réseau afin de mettre en évidence les failles de notre système.

Nous aurions également pu compléter l'interface web Zabbix afin d'afficher plus d'informations utiles dans les tableaux de bord de chaque appareil (serveur, routeur, switch) et d'améliorer l'expérience utilisateur.

Pour accéder à la translation d'adresse, une access-list a été créée sur nos routeurs EDGE.

Pour filtrer le trafic sur votre routeur REMFW, nous voulions configurer des access-list. Celles-ci fonctionnent sur la même base que les règles de pare-feux. Celles-ci sont hiérarchiques et permettent d'autoriser ou refuser les paquets venant de certaines sources en fonction de paramètres (port, protocole etc…). Nous n'avons pas pu les mettre en place dans le temps imparti, mettre en place du filtrage aurait pu bloquer une partie du réseau, de ce fait, nous avons privilégié le fonctionnel.

7. Conclusion

Cette SAÉ a mis à l'épreuve nos compétences en réseaux, systèmes et sécurité en nous permettant d'approfondir un certain nombre de notions vues sur l'ensemble de notre formation. Ces compétences ont pu être appliquées à un projet semblable au monde professionnel, de ce fait, nous pourrons capitaliser de l'expérience qui nous servira pour nos futurs projets ou dans notre vie professionnelle.

Initialement, le projet semblait difficile à réaliser dans son intégralité dès le départ, néanmoins nous avons pu concevoir une bonne partie de l'infrastructure et valider une partie de son fonctionnement. Même si nous aurions souhaité finaliser le projet dans son ensemble.

La partie la plus difficile de cette SAÉ était de s'approprier le sujet, en effet, du fait de sa complexité et de ces nombreuses consignes, il n'était pas évident de comprendre l'ensemble des attendus et de les réaliser. Nous avons dû passer du temps pour comprendre et débattre avec le groupe sur toute l'infrastructure et son fonctionnement.

Nous avions une bonne connaissance globale de l'ensemble des configurations à réaliser, mais le fait de les regrouper ensemble a mis en avant un certain nombre de points de blocages sur lesquels nous avons pu nous former pour monter en compétences.

8. Annexe

Plan d'adressage IP :
docs.google.com/spreadsheets/d/1UrzyShIVM7BfrDkqwa2btf4iJNKotk8z-yesrHXpPKQ/edit?usp=sharing

Dépôt Github de la Saé :
github.com/ThomasM2568/SAE501

Configurations des routeurs et switchs :
github.com/ThomasM2568/SAE501/blob/main/Configuration/Cisco/Routeur/OK/

Playbook Ansible :
github.com/ThomasM2568/SAE501/tree/main/Scripts/Ansible/Version_finale

Trello :
trello.com/b/rSSgP4aM/sa%C3%A9-501

Lucidchart :
lucid.app/lucidchart/4a432990-9e81-468c-a5c8-b203deb85ec6/edit?viewport_loc=-11%2C-11%2C2219%2C1047%2C0_0&invitationId=inv_07be847f-b0ae-49cf-84ec-0f6d5ed66c37